U.S.A. $3.95 CANADA $4.95
JULY 1989
VOLUME 8, NUMBER 3

ix
xclusive
rograms



ATARI GLYPHICS!

Ancient Egypt
Calendar-Maker



Red Squares:
Brain-Game Hit
Goes 8-Bit

VCR timer
now prints labels

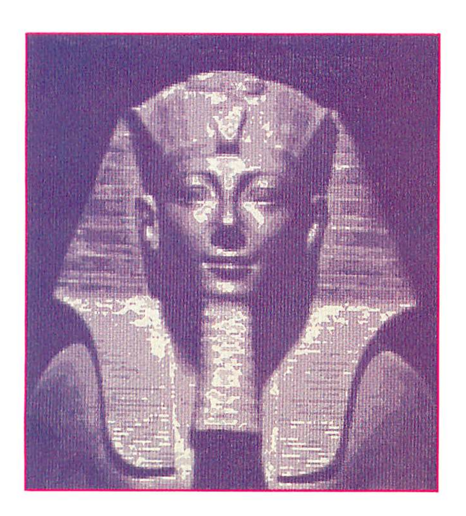






FEATURES

JULY 1989, VOL. 8, NO. 3

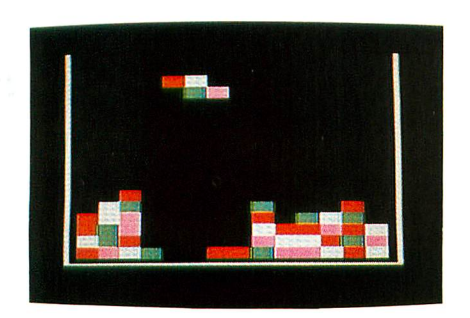


Egypt Calendar:
Page 20



Page 25

DEPARTMENTS



Red Squares:
Page 24

SOFTWARE LIBRARY 5 Easy-To-Type 8-Bit Listings

Antic—The Atari Resource (ISSN 0745-2527) is published monthly by Antic Publishing. Editorial offices are located at 544 Second Street, San Francisco, CA 94107. ISSN 0745-2527. Second Class Postage paid at San Francisco, California and additional mailing offices. POSTMASTER: Send address change to **Antic**, The Atari Resource, P.O. Box 1569, Martinez, CA 94553. Subscriptions: One year (12 issues) $28. Canada and Mexico add $8, other foreign add $12. Disk Edition (12 issues with disks) $59.95, all foreign add $25. (California residents add 6½ % sales tax for disk subscriptions. Editorial submissions should include text and program listings on disk and paper. Submissions will be returned if stamped, self-addressed mailer is supplied. **Antic** assumes no responsibility for unsolicited editorial material.



James Capparell
Publisher
John Taggart
Associate Publisher

EDITORIAL
Nat Friedland
Editor
Charles Jackson
Technical and Online Editor
Carolyn Cushman
Assistant Editor
Marta Deike
Editorial Coordinator

ART
Kathleen McKeown
Creative Services Director
Jim Warner
Art Director
Dwight Been
Associate Art Director
Georgia Solkov
Photo Editor and Cover Photography
Julianne Ososke
Production Manager
Kate Murphy
Advertising Production Coordinator
Jim Green
Technical Assistance for Cover Screen Shot

CIRCULATION
Manny Sawit
Director
Amber Lewis
Subscription Coordinator
Ann Dunlap
Retail Sales
Dennis Swan
Distribution Coordinator
(415) 957-0886

ADVERTISING
Marketing Manager
Diane Van Arsdall
Eastern Sales Representatives
Denny Riley
Western Sales Representative
Austin Holian
(415) 957-0886

ANTIC PUBLISHING, INC.
James Capparell
President and Chairman of the Board
Donald F. Richard
Richard D. Capparella
Directors
Lee Isgur
Advisor to the Board
John Taggart
Vice President
John Cady
Controller

GENERAL OFFICES
(415) 957-0886
544 Second Street, San Francisco, CA 94107

Credit Card Subscription and Catalog Orders
(800) 234-7001, Visa or MasterCard Only

SUBSCRIPTION CUSTOMER SERVICE
(415) 372-6002
Antic, P.O. Box 1569, Martinez, CA 94553

MINIMON FIX

The program *MiniMon* (**Antic**, April 1989), will not run on my Atari 1200XL as written. For an easy fix, change Line 1140 in Listing #1, MINIMON.BAS. The seven **boldfaced** numbers below are the only ones to change:

```
1140 DATA 2040961322050320662
4220116014400207312816200014218 20
021642050961322050322452411642O5
096125155
```

Now MINIMON.BAS will create a machine language file called MINIMON.EXE that will work on an 800 or 1200XL. The new version will not work an an XL/XE without a translator.

Paul Alhart
Lompoc, CA

***Antic** doesn't have a working 1200XL to check this on, but Paul Alhart has published a number of programs and Tech Tips in this magazine.* — ANTIC ED

DO 8-BITS SWIM?

I am presently the meet manager of our local swim club. Our club just yesterday finished hosting the Provincial Championships and I found the paperwork very time-consuming. I'd like to find a program I can use on my 130XE, one that will store information on the swimmers, seed the swimmers, produce time cards, and print the information in a program format. The only such program I can find is for IBM PCs. Is there one I can use on my Atari 8-bit?

Jerry Parsons
Gander, Newfoundland,
Canada

A spreadsheet program such as SynCalc could hold the information on swimmers, do some mathematical figuring for you, and print reports of the information. Talk to members of your local users group about available spreadsheets—or even finding a BASIC programmer who could write a custom program designed specifically for your set-up. — ANTIC ED.

BEGINNER'S BLUES

Sometimes I feel like a man sitting in a well-equipped garage feeling frustrated because I don't know how to use the tools. That's my situation with these cotton-pickin' computers and magazines I've picked up.

Don't get me wrong. . .since getting an Atari XE Game System for Christmas in '87 there's been a lot of time spent at the keyboard—but the potential is so much greater than the performance. I started reading ***Antic*** in early 1988, but I need the elementary stuff to lead the way, and it's been hard to find. I would like to know more about the different BASICs, where I can find a small business inventory program, how to get a word processor that fits me.

I hope you're thinking of us newcomers who aren't in school any longer. Give us a helping hand so we can catch up with you.

Bob White
Ferndale, MI

It's always hard to cover the needs of all our readers. Many of the topics you're interested in have been covered in previous issues, and most back issues are still available. If you don't know what issue you want, the ANTIC ONLINE Index on CompuServe is the most complete resource we can offer. You can search for articles, reviews and programs by title, subject, date and author. Very often, the complete text of the article is included in the index itself.

*Users groups are also an excellent source of help, and there are several active users groups in Michigan who produce a large joint newsletter, the **Atari Interface Magazine**. You can write them at Unicorn Publications, User Group Information, 3487 Braeburn Circle, Ann Arbor, MI 48108. Send them your address, phone number, and the kind of computer you own, so they can link you up with the appropriate group.*—ANTIC ED

continued on next page

DOS, 800 & XF551

It was with great sadness that I read your article on DOS-XE (March, 1989). I eagerly awaited Atari's new DOS, only to find out that it would not work with my old reliable 800. Playing around with the cartridge door switch so I can use SpartaDOS X does not appeal to me, either.

The Antic Arcade seems to say that SuperDOS 5.0 is compatible with all Atari 8-bit computers. Will it let me take advantage of all the XF551's capabilities?

Thomas Andrews
Manlius, NY

According to Arcade Manager Charles Cherry, SuperDOS 5.0 does work with the 800 and will give you full control of the XF551. — ANTIC ED

ATARIAN FRIENDS

I am the computer coordinator in the Lower School at the Wilmington Friends School. Fortunately, eight years ago the very wise principal of our school purchased four Atari 800 computers. Since then we have designed a curriculum around computer programming and word processing. The Atari computer is so easy for young children to program in graphic colors that we begin first graders designing and programming their own patterns. We continue programming through the fourth grade, with students strengthening their ability to plan a project.

Presently we have four 800s, two 800XLs, and 23 65XE computers. With the reasonable price of the system, many parents have also been able to get Ataris for their homes. I have prepared many lesson plans for teachers to use with BASIC on the Atari. It distresses me to see Atari systems take the back seat in conferences, catalogues, and everyday conversation.

Bertie Toler
Wilmington Friends School
101 School Road
Wilmington, DE 19803

CELEBRITY REVIEW

In my review of **Celebrity Cookbook**, published in the February 1989 **Antic**, I noted two problems with the program—quirky joystick response, and an inability to print the recipes, despite a very generic printer driver.

Well, I wrote the company, and they did send me a fresh copy that fixed these bugs. It was six weeks in arriving, perhaps due to having moved their offices from California to Maryland about that time. But the support *was* there for me, so I wouldn't hesitate to recommend the product.

David Merrihue
Daly City, CA

Celebrity Cookbook ($29.95) is available from U.S.A. Media, 7810 Malcolm Road, Clinton MD 20735. (301) 868-5494. — ANTIC ED

FROM DEBUGGING TO BUG SPRAY

I would like to start an Atari Farmer's and Gardener's user group. Anyone who uses an Atari 8-bit to help them with their gardening or farming is welcome to join. I would like to issue a disk full of useful programs, if we can accumulate enough. We are particularly interested in artificial intelligence applications for the purpose of sorting out plant nutrient requirements, programs to track nutrient usage, or anything else that would help with the task of growing food. This includes hardware interfacing with real world sensors, etc.

Anyone interested should drop us a letter with a self-addressed stamped envelope, and we'll let you know how it's going. The Atari 8-bit is the most cost-effective computer around. Let's get on the ball and see if we can apply it to the much needed job of producing wholesome food.

GROMOR-SYSTEMS
Lee Jones
Rt. 1 Box 76-B
Pleasantville, TN
37147-9801

ATARIWRITER DRIVERS

I have a problem with the subscript and superscript commands in AtariWriter. The printer goes into subscript or superscript mode, but won't come out, so everything is printed in tiny print slightly above or below the normal print line. I have an Epson LX-80 printer. Is there anything I can do to fix the problem?

K. Helton
Sacramento, CA

*The Antic Arcade's **Printer Driver Construction Set** (AP0131, $19.95) will let you set up a special driver file that will "make your AtariWriter cartridge compatible with any printer."* — ANTIC ED

DRIVE NEEDS DOS

We have an Atari 800XL and a disk drive, and are thinking of subscribing to your disk magazine. Do you need DOS to play the disks?

J.E. Barclay
Lake Havasu City, AZ

DOS stands for Disk Operating System, and as the name suggests, you need some sort of DOS to use a disk drive. Fortunately, the Antic Monthly Disk always comes with Atari DOS 2.0 on it—all you need to do is put the disk in the drive and turn the computer on, and the disk menu will appear. — ANTIC ED

Antic welcomes your feedback, but we regret that the large volume of mail makes it impossible for the Editors to reply to everyone. Although we do respond to as much reader correspondence as time permits, our highest priority must be to publish I/O answers to questions that are meaningful to a substantial number of readers.

Send letters to: Antic I/O Board, 544 Second Street, San Francisco, CA 94107.



COMPUTER SOFTWARE SERVICES

ATARI

P.O. BOX 17660, ROCHESTER, N.Y. 14617

PHONE (716) 467-9326

NOW "BEST BUY" RATED!

JUST RELEASED!

$99.95 "SUPER ARCHIVER II"!

(for ATARI 1050 drives)

NOW! COPYS all **ENHANCED DENSITY** programs plus retains all of the features of our World Famous **SUPER ARCHIVER!** (see below). Allows you to **COPY** or **CREATE** single or **ENHANCED** density protection schemes (including **PHANTOM SECTORS!**). Completely automatic; compatible with the **BIT-WRITER!**; the **ULTIMATE BACKUP/PROGRAMMING** device! Only $99.95 plus $4 S/H/I **NOTICE!** If you already own a **SUPER ARCHIVER!**, you can upgrade to a **SUPER** ARCHIVER II for only $29.95 plus $4 S/H/I (disk only - no additional hardware required).

THE

$69.95 "SUPER ARCHIVER"!® $69.95

(for ATARI 1050 drives)

The new **SUPER ARCHIVER**, obsoletes **all copying devices** currently available for the **ATARI 1050! It eliminates** the need for Patches, **PDB files**, Computer Hardware, etc. Copies are **exact duplicates** of originals and **will run on any drive**; without exaggeration, the **SUPER ARCHIVER** is the **most powerful PROGRAMMING/COPYING** device available for the 1050! Installation consists of a plug-in chip and 6 **simple** solder connections. Softwares included. Features are:

- **TRUE DOUBLE DENSITY**
- **ULTRA-SPEED** read/write
- **FULLY AUTOMATIC COPYING**
- **SUPPORTS EXTRA MEMORY**
- **SCREEN DUMP** to printer
- **TOGGLE HEX/DEC DISPLAY**
- **SECTOR** or **TRACK TRACING**
- **AUTOMATIC DIAGNOSTICS**
- **DISPLAYS HIDDEN PROTECTION**
- **ADJUSTABLE/CUSTOM SKEWING**
- **AUTOMATIC SPEED COMPENSATION**
- **AUTOMATIC/PROGRAMMABLE PHANTOM SECTOR MAKER**
- **ARCHIVER/HAPPY ARCHIVER COMPATIBLE**
- **BUILT-IN EDITOR** - reads, writes, displays up to 35 sectors/track short
- **BUILT-IN CUSTOM FORMATTER** up to 40 sectors/track
- **BUILT-IN DISASSEMBLER**
- **BUILT-IN MAPPER** - up to 42 sectors/track
- **DISPLAYS/COPIES** Double Density **HEADERS**
- **AUTOMATIC FORMAT LENGTH CORRECTION**
- **SIMPLE INSTALLATION**

The **SUPER ARCHIVER** is so **POWERFUL** that only programs we know that can't be copied are the newer **ELECTRONIC ARTS** and **SYNFILE/SYNCALC** (34 FULL sectors/track). If you want it **ALL** . . . buy the **"BIT-WRITER"!** also . . . then you'll be able to copy even these programs! Only $69.95 plus $4 S/H/I.

$79.95 "BIT-WRITER"! $79.95

The Super Archiver **"BIT-WRITER"!** is capable of **duplicating even** the "uncopyable" **EA** and **SYN** series which employ 34 FULL sectors/tracks. **"BIT-WRITER"!** is capable of reproducing these and **FUTURE** protection schemes of non physicially damaged disks. **PLUG-IN** circuit boards and 4 **simple** solder connections. The **"SUPER ARCHIVER** with **"BIT-WRITER"!** is the ultimate **PROGRAMMING/COPYING** device for Atari 1050's. **EXACT DUPLICATES** of originals are made! Copies run on **ANY** drive. Must be used with Super Archiver. Only $79.95 plus $4 S/H/I.

$69.95 "ULTRA SPEED PLUS" $69.95

Imagine a universal XL/XE Operating System so easy to use that anyone can operate it instantly, yet so versatile and powerful that every Hacker, Programmer and Ramdisk owner will wonder how they ever got along without it! Ultra Speed Plus puts unbelievable speed and convenience at your fingertips. Use **ANY DOS** to place an **ULTRA SPEED** format on your disks, boot any drive (1-9) upon power-up, format your **RAMDISK** in Double Density, activate a built-in **400/800 OS** for software compatibility, plus dozens of other features too numerous to mention! Below are just a **FEW** features you'll find in the amazing OS:

- **ULTRA** Speed SI0 for most modified drives
- **ULTRA** Speed is toggleable
- Boot directly from **RAMDISK**
- Special timer circuits not required for 1 or 2 Meg upgrades
- Background colors adjustble
- Reverse use of **OPTION** key
- Cold-start without memory loss
- Built-in floppy/disk configuration editor (1-9)
- Built in **RAMDISK** configuration editor (1-9)
- **RAMDISK** exactly duplicates floppy drive so sector copying and sector editing are now possible
- Built in **MINI** Sector Copier
- Toggle **SCREEN OFF** for up to 40% increase of processing speed.
- Toggle internal **BASIC**
- Ram resident disk loader program **(MACH 10 menu)**
- **DOUBLE DENSITY RAMDISK** capable
- Entire **MEMORY** test that pinpoints defective **RAM** chip
- Boot any drive (1-9) upon power-up or cold-start
- Supports memory upgrades up to TWO MEGABYTES
- **THREE** Operating Systems in one **(XL/XE, 400/800, ULTRA SPEED PLUS)**

$29.95 "XF551 ENHANCER!" $29.95

The XF551 Atari drive is a fine product with one major flaw...it writes to side TWO of your flopping disks **BACKWARDS**. This causes read/write incompatibility problems with all other single sides drives made for Atari such as Indus. Trak, Rana, Percom, Astra, Atari 1050, Atari 810, etc. Add the **XF551 ENHANCER** to the new XF551 drive and your problems are over! This device will restore 100% compatibility while retaining original design qualities of Atari's super new drive. The **XF551 ENHANCER** is a **MUST** for all XF551 Owners. Installation is simple. Only $29.95 plus $4 S/H/I.

DEALER/DISTRIBUTOR/USER GROUP Discount availalbe call for info.
PHONE Orders - **MASTER CARD, VISA MAIL** - Money Orders, Checks

ATARI

*SHIPPING! Add $4 for Shipping/Handling/Insurance within the U.S.A. UPS BLUE LABLE (2nd DAY AIR) available for $3 extra per shipment.

CALL TODAY! (716) 467-9326
9 am - 5 pm (EST) WEEKDAYS

COMPUTER SOFTWARE SERVICES
P.O. Box 17660
Rochester, N.Y. 14617

$39.95 RICHMAN'S $39.95

80 Column Word-Processor!

Easy to use, very **powerful**, and **NO ADDITIONAL HARDWARE** required! Works with TV or Monitor! This **"DISK ONLY"** 80 Column Word-Processor is **simple to use** while offering numerous features such as:

- Completely Menu driven
- Self contained DOS
- Complete Text formatting
- Built in Printer Drivers
- Page numbering
- Justify and Endfill
- High Speed Cursor
- Dictionary
- 80 COLUMNS ON THE SCREEN!
- Insert and Delete
- Search Mode
- View only mode
- Memory Statistics
- Alternate Output
- Change screen Colors, borders, luminesce
- Mouse compatible

One expert has compared **40 domestic** and **foreign word-processors** and concluded that the **RICHMANS 80 Column Word-Processor is the best!** Only $39.95 plus $4 S/H/I. (64K required).

$39.95 THE "QUINTOPUS!" $39.95

The **"QUINTOPUS!"** is an inexpensive device that provides a tremendous amount of convenience while eliminating the problems associated with the endless "daisy-chaining" of peripherals (eg: drives, interfaces, printers, modems, cassettes, etc.). The **"QUINTOPUS!"** is an I/O port expander that converts a single I/O output/input into five additional outputs/inputs. Instead of daisy-chaining all of your peripherals (which often times results in passing a signal through 30 feet or more of cable!), the **"QUINTOPUS!"** allows each device to be connected directly to the computers through only three or four feet of cable. This is particularly useful when attempting to use ULTRASPEED or WARPSPEED data transfer rates. The **"QUINTOPUS!"** also provides the "extra" I/O ports often needed to connect devices not having daisy-chaining capabilities. Cable resistance and compacitance are greatly reduced thereby significantly improving the opportunity of accurate data transfers! Only $39.95 plus $4 S/H/I.

$59.95 THE "QUINTOPUS!" $59.95

(with SWITCHABLE PORTS!)

This Deluxe version of the **"QUINTOPUS!"** has all of the above features with the additional benefits of **two SWITCHABLE PORTS!** This means you can connect two computers to one printer or two printers to one computer; you can switch in a computer/printer combination and while you're printing out a long document, switch in a second computer to access a modem, disk drive, cassette deck, etc! Switch multiple combinations of peripherals or computers without the hassel or re-arranging I/O cables . . . . simply flip a switch! Only $59.95 plus $4 S/H/I.

LIMITED SPECIALS!

Previously we listed these products under our **FIRESALE** ad but many of our customers feared that the software items might be smoke or water damaged. They're NOT! All of the items listed below are BRAND NEW...only their literature or packing may have slight cosmetic imperfections. All items are fully warrantied! Quantities are limited! No rainchecks!

	NORMALLY	LIMITED SPECIALS
1. PILL (without case)	$ 70	$ 19
2. SUPER PILL (without case)	80	22
3. XL FIX ROM	70	19
4. ULTRA MENU/DOS	30	19
5. DISKCRACKER (Newest version)	50	19
6. ELECTRONIC PHANTOM SECTOR MAKER DELUXE	60	35
7. RICHMANS 80 COLUMN WORD PROCESSOR	60	29
8. MIRACLE (Disk only Version of Impossible)	70	19
9. IMPOSSIBLE for 800 or 800XL	150	69
10. XL MATE	30	15
11. COMPACTOR	30	15
12. KLONE II (Generic HAPPY backup)	100	75
13. SILENCER	30	19
14. BLACK PATCH (MASTER)	50	25
15. BLACK PATCH DATA DISKS 1 & 2	20	10

*Order 5 items or more and we will pay the freight!

These SPECIALS are for C.S.S. customers only - NO DEALER DISCOUNTS - NO RAINCHECKS - LIMITED QUANTITIES

Machine Language Stringer

Save 7 seconds on each BASIC subroutine.

By Andy Barton

Machine Language Stringer takes machine language object code and converts it into string format for use in your own BASIC programs. This BASIC program works on Atari 8-bit computers with at least 48K memory and disk drive.

Most machine language routines in BASIC programs are in the form of string data. This is done partly because strings take up less space than numerical DATA statements. They frequently don't need to be POKEd into a memory location.

When I tried translating a machine language routine in one of my programs into string format, I discovered the best reason for using strings. The string format virtually eliminated the seven seconds used to READ the 144 bytes of data and POKE them onto page 6. I was sold. I dearly hate to wait for slow computers.

I developed Machine Language Stringer to do the near-impossible manual task of taking the object code file of a machine language program (the executable code) and converting it into a set of BASIC program lines that will produce the proper string data.

GETTING STARTED

Type in Listing 1, STRDAT.BAS, and check it with TYPO II. Be sure to SAVE a copy to disk before you RUN it.

If you have difficulty typing the special characters in Line 460, don't type them in. Instead, type Listing 2, check it with TYPO II and SAVE a copy. When you RUN Listing 2, it creates these hard-to-type lines and stores them in a file called LINES.LST.

To merge the two programs, LOAD "D:STRDAT.BAS" and then ENTER "D:LINES.LST." Remember to SAVE the completed program before you RUN it.

When you RUN the program, you will be asked for the object file name. If you forget the "D:" or the name is not found you will be asked again. Next you are asked for a starting line number. Be sure you choose one that will not overlap lines in your BASIC program or in this one.

Finally, you are asked for a name for the machine language string (maximum of 2 characters). The program will add a numerical extender to this name, starting with 1. The program will now go about the business of reading the object file and building the BASIC line(s) that will be incorporated into this program using Atari's forced read mode.

When the program is done, there are three more steps for you to take to incorporate the string into your basic program.

1. LIST the new lines to a disk/cassette file, for example:

LIST "D1:YOURPRG.STR",1000,1005.

2. LOAD your BASIC program and ENTER the string data file, for

example:

ENTER "D1:YOURPRG.STR".

3. Make a USR comand to run the ML string.

PROGRAM NOTES

There are two numbers that cannot be displayed in a string—34 and 155. 34 is ASCII for a quotation mark and 155 is ASCII for a carriage return (return key). This program handles this problem by creating a separate line that inserts the number into its proper place in the string, for example:

1001 ML1$(72,72)=CHR$(155).

There are two types of machine language programs, ones that are fully relocatable and ones that are fixed at a particular memory location. Jump (JMP) and jump subroutine (JSR) comands use absolute rather than relative addressing and thus require the program to be at the specific location to which it was assembled.

Machine Language Stringer accommodates this by creating a final BASIC line which provides a brief machine language string to move the string data to the memory location indicated by the object file, for example:

1006 X=USR(ADR("hh. . ."), FROM , TO , NO. BYTES)

This line is provided regardless of which type of machine language program you wrote. If your program is fully relocatable, this line can be deleted.

It is possible to create a program that is loaded into two or more separate memory locations. For example, a section of subroutines could be fixed onto page 6 and the main program could be totally relocatable. Machine Language Stringer accomodates this by using the numeric extender mentioned above. Each time a new load address is indicated in the object file, the extender is increased by one, creating a new string name. Each string is provided with its own loader.

As mentioned above, for a program to be relocatable it cannot use absolute addressing with jump instructions. I have found no way around this problem with subroutines other then placing them on Page 6 or some other safe, fixed location.

However, there are two tricks I have discovered for JMP instructions. The problem arises when I would use a branch instruction, but find that its range (126 bytes) was too short so I would be forced to use a JMP instrution. The first crude but effective solution involves setting up intermediate branches to one or more areas within range, but outside the flow of the program. Here is an example:

```
LOOP   LDY #0
PART1  LDA ($D0),Y
       . . .
       BEQ PART1
       BNE PART2 ;BRANCHES
                 ;OVER
BP1    BCS LOOP  ;INTER-
                 ;MEDIATE
                 ;BRANCH
PART2  ASL A
       . . .
       SEC       ;SET CARRY
                 ;TO
       BCS BP1   ;FORCE A
                 ;BRANCH
```

The second solution is more versatile, using the indirect jump instruction JMP(XXXX). It involves passing the address of the relocateable ML string to the ML program in BASIC's USR command. The ML program then figures the relative distance from the start of the program to the targeted instruction, adds this to the starting address of the string and saves the results on page 6 for the JMP(XXXX) to use. Here are two examples, first in BASIC:

X=USR(ADR(ML$),ADR(ML$))

In ML, this would be:

```
IJP1 = $600 ;SAFE
            ;STORAGE
            ;FOR
IJP2 = $602 ;INDIRECT
            ;JUMP
            ;ADDRESS
START * = $5600
      PLA
      PLA ;HI BYTE OF
          ;ADDRESS OF ML
          ;STRING
      TAX
      PLA ;LOW BYTE OF
          ;ADDRESS
      TAY ;SAVE IF MORE
          ;THEN ONE JUMP
          ;TARGET
          ;NEEDED
      CLC
      ADC #<TARG1-START
          ;ADD LOW BYTE
          ;OF TARGET
          ;ADDRESS OFFSET
          ;TO ML STRING
          ;ADDRESS
      STA IJP1
      TXA
      ADC #>TARG1-START
          ;ADD HI BYTES
      STA IJP1+1
      TYA ;GET LOW BYTE
          ;STRING ADDRESS
      CLC ;FOR SECOND
          ;TARGET
      ADC #<TARG2-START
      . . . ;etc.
TARG1 SEC ;SOMEWHERE IN
          ;MAIN PROG.
      . . .
      . . .
      JMP(IJP1)
```

You may have to modify the program to get it to work with character set files. This program strips the first two control characters from the file, so you would end up with 1022 instead of 1024 bytes in your character set files.

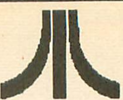
As always, whenever modifying your programs you should first make backups of the originals, in case problems arise.

Andy Barton has been a regular contributor to Antic since 1984. His machine language game, Exwall, is this month's Super Disk Bonus.

Listing on page 37

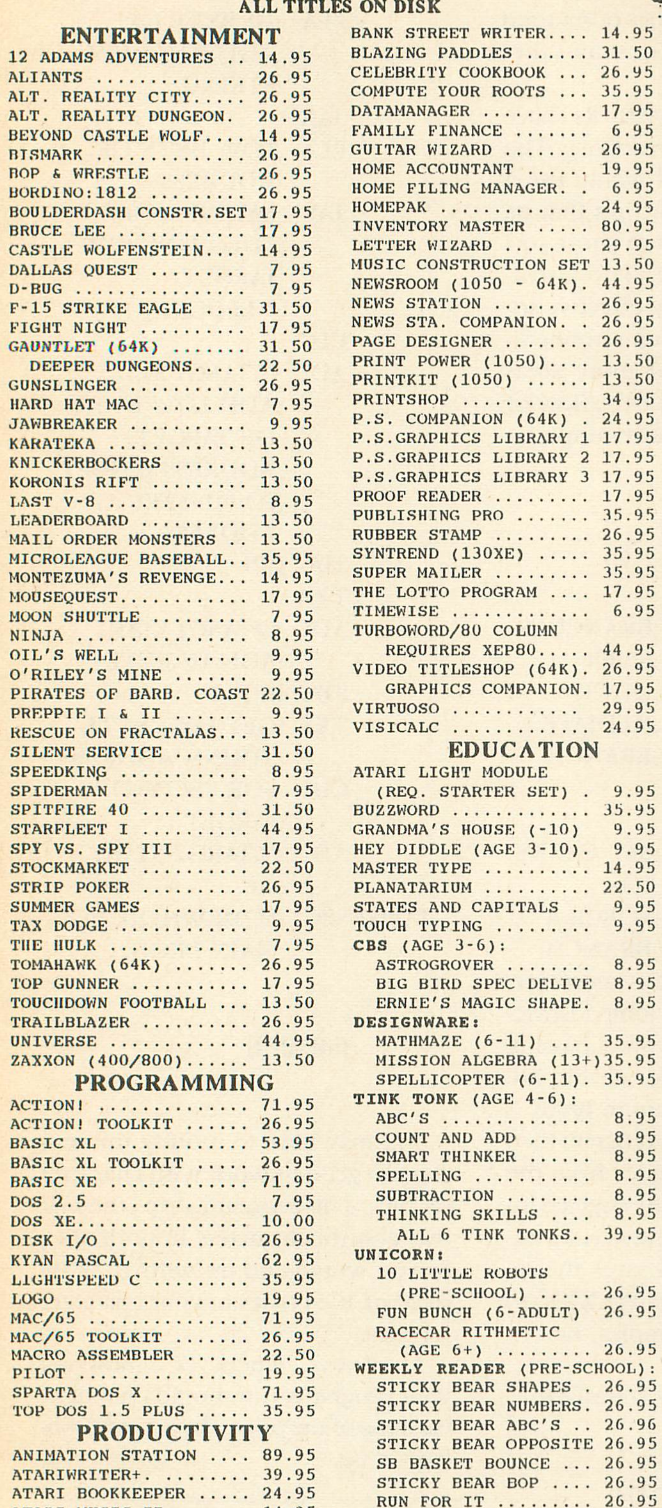
B&C ComputerVisions

3257 KIFER ROAD
SANTA CLARA, CA 95051
(408) 749-1003
(408) 749-9389 FAX



STORE HOURS
TUE - FRI 10am - 6pm
SAT - 10am - 5pm
CLOSED SUN - MON

800/XL/XE SOFTWARE
ALL TITLES ON DISK



ENTERTAINMENT

12 ADAMS ADVENTURES	14.95
ALIANTS	26.95
ALT. REALITY CITY	26.95
ALT. REALITY DUNGEON	26.95
BEYOND CASTLE WOLF	14.95
BISMARK	26.95
BOP & WRESTLE	26.95
BORDINO:1812	26.95
BOULDERDASH CONSTR.SET	17.95
BRUCE LEE	17.95
CASTLE WOLFENSTEIN	14.95
DALLAS QUEST	7.95
D-BUG	7.95
F-15 STRIKE EAGLE	31.50
FIGHT NIGHT	17.95
GAUNTLET (64K)	31.50
DEEPER DUNGEONS	22.50
GUNSLINGER	26.95
HARD HAT MAC	7.95
JAWBREAKER	9.95
KARATEKA	13.50
KNICKERBOCKERS	13.50
KORONIS RIFT	13.50
LAST V-8	8.95
LEADERBOARD	13.50
MAIL ORDER MONSTERS	13.50
MICROLEAGUE BASEBALL	35.95
MONTEZUMA'S REVENGE	14.95
MOUSEQUEST	17.95
MOON SHUTTLE	7.95
NINJA	8.95
OIL'S WELL	9.95
O'RILEY'S MINE	9.95
PIRATES OF BARB. COAST	22.50
PREPPIE I & II	9.95
RESCUE ON FRACTALAS	13.50
SILENT SERVICE	31.50
SPEEDKING	8.95
SPIDERMAN	7.95
SPITFIRE 40	31.50
STARFLEET I	44.95
SPY VS. SPY III	17.95
STOCKMARKET	22.50
STRIP POKER	26.95
SUMMER GAMES	17.95
TAX DODGE	9.95
THE HULK	7.95
TOMAHAWK (64K)	26.95
TOP GUNNER	17.95
TOUCHDOWN FOOTBALL	13.50
TRAILBLAZER	26.95
UNIVERSE	44.95
ZAXXON (400/800)	13.50

PROGRAMMING

ACTION!	71.95
ACTION! TOOLKIT	26.95
BASIC XL	53.95
BASIC XL TOOLKIT	26.95
BASIC XE	71.95
DOS 2.5	7.95
DOS XE	10.00
DISK I/O	26.95
KYAN PASCAL	62.95
LIGHTSPEED C	35.95
LOGO	19.95
MAC/65	71.95
MAC/65 TOOLKIT	26.95
MACRO ASSEMBLER	22.50
PILOT	19.95
SPARTA DOS X	71.95
TOP DOS 1.5 PLUS	35.95

PRODUCTIVITY

ANIMATION STATION	89.95
ATARIWRITER+	39.95
ATARI BOOKKEEPER	24.95
ATARI MUSIC II	14.95
AWARDWARE (1050)	13.50
BANK STREET WRITER	14.95
BLAZING PADDLES	31.50
CELEBRITY COOKBOOK	26.95
COMPUTE YOUR ROOTS	35.95
DATAMANAGER	17.95
FAMILY FINANCE	6.95
GUITAR WIZARD	26.95
HOME ACCOUNTANT	19.95
HOME FILING MANAGER	6.95
HOMEPAK	24.95
INVENTORY MASTER	80.95
LETTER WIZARD	29.95
MUSIC CONSTRUCTION SET	13.50
NEWSROOM (1050 - 64K)	44.95
NEWS STATION	26.95
NEWS STA. COMPANION	26.95
PAGE DESIGNER	26.95
PRINT POWER (1050)	13.50
PRINTKIT (1050)	13.50
PRINTSHOP	34.95
P.S. COMPANION (64K)	24.95
P.S.GRAPHICS LIBRARY 1	17.95
P.S.GRAPHICS LIBRARY 2	17.95
P.S.GRAPHICS LIBRARY 3	17.95
PROOF READER	17.95
PUBLISHING PRO	35.95
RUBBER STAMP	26.95
SYNTREND (130XE)	35.95
SUPER MAILER	35.95
THE LOTTO PROGRAM	17.95
TIMEWISE	6.95
TURBOWORD/80 COLUMN REQUIRES XEP80	44.95
VIDEO TITLESHOP (64K)	26.95
GRAPHICS COMPANION	17.95
VIRTUOSO	29.95
VISICALC	24.95

EDUCATION

ATARI LIGHT MODULE (REQ. STARTER SET)	9.95
BUZZWORD	35.95
GRANDMA'S HOUSE (-10)	9.95
HEY DIDDLE (AGE 3-10)	9.95
MASTER TYPE	14.95
PLANATARIUM	22.50
STATES AND CAPITALS	9.95
TOUCH TYPING	9.95
CBS (AGE 3-6):	
ASTROGROVER	8.95
BIG BIRD SPEC DELIVE	8.95
ERNIE'S MAGIC SHAPE	8.95
DESIGNWARE:	
MATHMAZE (6-11)	35.95
MISSION ALGEBRA (13+)	35.95
SPELLICOPTER (6-11)	35.95
TINK TONK (AGE 4-6):	
ABC'S	8.95
COUNT AND ADD	8.95
SMART THINKER	8.95
SPELLING	8.95
SUBTRACTION	8.95
THINKING SKILLS	8.95
ALL 6 TINK TONKS	39.95
UNICORN:	
10 LITTLE ROBOTS (PRE-SCHOOL)	26.95
FUN BUNCH (6-ADULT)	26.95
RACECAR RITHMETIC (AGE 6+)	26.95
WEEKLY READER (PRE-SCHOOL):	
STICKY BEAR SHAPES	26.95
STICKY BEAR NUMBERS	26.95
STICKY BEAR ABC'S	26.96
STICKY BEAR OPPOSITE	26.95
SB BASKET BOUNCE	26.95
STICKY BEAR BOP	26.95
RUN FOR IT	26.95
PIC BUILDER	26.95



800/XL/XE SOFTWARE
ALL TITLES ON CARTRIDGE

ENTERTAINMENT

3D TIC-TAC-TOE	9.95
AIRBALL (XL/XE)	24.95
ALIEN AMBUSH	9.95
ACE OF ACES (XL/XE)	24.95
ARCHON	19.95
ASTEROIDS	15.95
ASTRO CHASE	14.95
ATARI TENNIS	9.95
ATLANTIS	14.95
BALL BLAZER	19.95
BARNYARD BLASTER (REQ. LIGHT GUN)	24.95
BATTLEZONE	19.95
B.C. QUEST FOR TIRES	19.95
BLUE MAX	19.95
BOULDERS & BOMBS	14.95
CAVERNS OF MARS	14.95
CENTIPEDE	14.95
CHICKEN	9.95
CHOPLIFTER	14.95
CLAIM JUMPER (400/800)	9.95
CLOUDBURST	9.95
CRIME BUSTER (REQ. LIGHT GUN)	24.95
CROSSBOW	24.95
CROSSFIRE	9.95
CRYSTAL CASTLES(XL/XE)	19.95
DARK CHAMBERS (XL/XE)	24.95
DAVIDS MIDNIGHT MAGIC	19.95
DEFENDER	14.95
DELUXE INVADERS	7.95
DESERT FALCON	19.95
DIG DUG	19.95
DONKEY KONG	5.00
DONKEY KONG JR.	19.95
EASTERN FRONT (1941)	19.95
E.T. PHONE HOME	9.95
FIGHT NIGHT	19.95
FINAL LEGACY	19.95
FOOD FIGHT (XL/XE)	19.95
FOOTBALL	14.95
FROGGER	14.95
GALAXIAN	9.95
GATO	24.95
GORF (400/800)	5.00
GYRUSS	14.95
HARDBALL	19.95
INTO THE EAGLES NEST	19.95
JOURNEY TO PLANETS	9.95
JOUST	19.95
JUNGLE HUNT	19.95
KABOOM!	14.95
KARATEKA	19.95
KRAZY ANTICS	14.95
LODE RUNNER	24.95
MARIO BROS.	19.95
MEGAMANIA	9.95
MILLIPEDE	14.95
MISSILE COMMAND	5.00
MOON PATROL	19.95
MR. COOL	9.95
MS. PAC MAN	19.95
NECROMANCER	19.95
ONE ON ONE (XL/XE)	19.95
PAC MAN	5.00
PENGO	19.95
POLE POSITION	19.95
POPEYE	14.95
Q-BERT	14.95
QIX	14.95
RESCUE ON FRACTALAS	19.95
RETURN OF THE JEDI	14.95
ROBOTRON:2084	19.95
SKY WRITER	14.95
SLIME (400/800)	9.95
SPACE INVADERS	14.95
STAR RAIDERS	5.00
STAR RAIDERS II	19.95
SUBMARINEE COMMANDER	14.95
SUMMER GAMES (XL/XE)	24.95
SUPER BREAKOUT	9.95
SUPER COBRA	14.95
THUNDERFOX	19.95
TRACK & FIELD	24.95
TURMOIL	9.95
WIZARD OF WOR	5.00

PRODUCTIVITY

ATARIWRITER	19.95
MICROFILER	22.50

EDUCATION

ATARILAB STARTER SET	29.95
FUN WITH ART	14.95
MATH ENCOUNTERS	9.95
FISHER PRICE (PRE SCHOOL):	
DANCE FANTASY	8.95
LINKING LOGIC	8.95
LOGIC LEVELS	8.95
MEMORY MANOR	8.95
SPINNAKER (AGE 3-10):	
ALF IN COLOR CAVES	9.95
ALPHABET ZOO	9.95
DELTA DRAWING	9.95
FACEMAKER	9.95
KIDS ON KEYS	9.95
KINDERCOMP	9.95
(AGE 7 - ADULT):	
ADV.CREATOR (400/800)	9.95
FRACTION FEVER	9.95

SUPER SPECIALS
RECONDITIONED ATARI MERCHANDISE
30 DAY WARRANTY

800 (48K) COMPUTER $79.95	**SPACE AGE JOYSTICK $5.00.**	**1030 MODEM WITH EXPRESS! $24.95**
400 (16K) COMPUTER $29.95	**ATARI TRACKBALL $9.95**	**1010 PROGRAM RECORDER $29.95**
1020 COLOR PRINTER/PLOTTER $19.95 40 COLUMNS WIDE (new in box)	**ATARI BOOKKEEPER $14.95 - NO BOX**	**DISKETTES AS LOW AS 20 CENTS** 10 FOR $4.00 / 100 FOR $29.95 / 1000 FOR $200 / MOST ARE UNNOTCHED WITH OLD SOFTWARE
	ATARI NUMERIC KEYPAD $7.95	

SHIPPING INFORMATION - Prices do not include shipping and handling. Add $5.00 for small items ($8.00 Min. for Canada). Add $8.00 for disk drive. Add $2.75 for C.O.D. Calif. res. include 7% sales tax. Mastercard and Visa accepted if your telephone is listed in your local phone directory. Orders may be pre-paid with money order, cashier check, or personal check. Personal checks are held for three weeks before order is processed. C.O.D orders are shipped via UPS and must be paid with cash, cashier check or money order. International and APO orders must be pre-paid with cashier check or money order. $20.00 minimum on all orders. All sales are final - no refunds - prices are subject to change.

Phone orders accepted **TUESDAY THROUGH FRIDAY** from 10:00 am to 6:00 pm PST.

We carry a full line of **ATARI** products - large public domain library - write or call for **free** catalogue

PRICES SUBJECT TO CHANGE WITHOUT NOTICE - ALL SALES ARE FINAL

Customizing the Atari Operating System Device Handlers:



By Bob Martin & Martin Mercorelli

The final half of this series is for experienced MAC/65 programmers. It describes an interactive handler that saves machine language programs as boot files. This program requires MAC/65 and OS/A+, and works on all 8-bit Atari computers of any memory size, with disk drive.

In the previous month's **Antic**, the first half of this series introduced device handlers and described how they work by creating two simple handlers. This final installment describes the creation and operation of MAKEBOOT, a more sophisticated device handler.

The MAKEBOOT handler lets you save object code as a boot file and convert binary load files to boot files. Before the MAKEBOOT handler can do this, however, it does something quite unique—it asks you questions.

Although almost every useful computer program prompts you for information, handlers do not. Since the CIO (Central Input/Output) uses device handlers whenever it operates, the device handlers cannot easily use CIO to prompt you for information—the CIO is busy.

If we try to use the CIO while it's busy, your Atari usually—but not always—becomes confused and acts strange. This is why handlers should not use CIO for I/O to the screen or

keyboard.

The catch is that we often want to interact with a program while a handler is in use. Therefore we must use the screen or keyboard handlers directly, *without* going through the CIO.

MAKEBOOT is an example of such a handler. The MAKEBOOT program requires you to direct the handler operation and make some decisions while it's operating. Your Atari operating system has a built-in mechanism for accomplishing this.

Uses for this program include loading an alternate program into the same area occupied by DOS, or initializing your Atari before DOS is loaded. For example, you could load in the modification to the printer handler. You could use this program to produce bootable games or programs that produce a disk menu.

BOOT FILES

A boot file is a machine language program which resides on the outermost sectors of a disk. It is automatically loaded whenever you boot with that disk. On the disk, the boot file is a continuous, uninterrupted file which begins in the first sector and occupies successive sectors until the end of the file. On disk, there are no breaks between the end of one sector and the beginning of the next—and no directory.

Binary load files are machine language programs which may reside anywhere else on the disk. On the disk, a binary load file may be broken into sector-sized pieces and scattered throughout the disk. The last three bytes of each sector direct your Atari to the next sector of the file. Your Atari treats these sector links as "Continued On Sector *xxx*" messages.

Every time you boot a disk, your Atari checks the first six bytes of the first sector to determine what action to take next.

Byte 0, the first of these six bytes, is used as a flag. (A zero in this location denotes a boot file.) This value is stored to DFLAGS, memory location 576 ($0240).

Byte 1 contains the number of sectors to load, bytes 2 and 3 tell your Atari where to load the boot data (this is the "load address"), and the next two bytes tell your Atari where to go after the program starting at byte 6 is executed (this is the "initialization address").

The program starting at byte 6 is an initialization program and usually ends with an RTS (ReTurn from Subroutine) instruction. If there is no initialization routine, then byte 6 *must* be an RTS instruction, which is represented by a 96 ($60).

If the initialization routine doesn't start at byte 6, then byte 6 must be a JuMP instruction, directing your Atari to the start of the initializing routine.

When the initialization program ends, the operating system jumps to the memory location given in bytes 4 and 5.

The program in Listing 1, MAKEBOOT.M65, treats the first nine bytes of sector 1 as if it was structured as shown in *Figure 1.* This structure requires nine bytes of data on sector 1.

When a series of sectors is loaded as part of the initial boot, the sectors are loaded sequentially in memory. For example, if the initial load address is 1000, then sector 1 loads its data starting at 1000 (including the first six bytes), sector 2 loads its data starting at 1128 (there are 128 bytes per sector), etc.

This initial boot sector load is called the first-stage load. For a DOS format disk, three sectors are loaded in the first-stage load, then your Atari loads DOS.SYS, a second-stage load.

Since the boot sectors are loaded in memory sequentially, the specified load address (bytes 2 and 3) is treated as the memory location of byte 0 of sector 1, and data on the disk is calculated relative to that address. In the example, if the load address was 1000, then byte 0 of sector 1 corresponds to memory address 1000, byte 0 of sector 2 corresponds to address 1128, etc. Note that the initialization routine of the boot sectors starts at 1006, since all of sector 1 (including the first six bytes) is loaded.

The MAKEBOOT handler does something quite unique—it asks you questions.

THE LOAD FILE

Now that we've determined a way to put data on the boot disk, we need to know how data will be received from the CIO. Load files are loaded by DOS as a series of data blocks. A block can be any length, but they're typically 251 bytes long (at least in MAC/65) and preceded by two two-byte numbers. The first number is the starting address, where the first byte is stored. The second number is the ending address, where the last byte is stored.

If DOS was loading this file in memory, each byte of the block would be stored sequentially until the ending address was reached. Then it would repeat the process until all the data had been loaded.

This varies only at the start of a file and when appending files. The start of a file has two bytes of the value 255 that identify it as a load file. When one load file is appended to another, these bytes are carried over to the load data. This means that a data block is preceded by either four or six bytes,

where the first two are 255, 255.

Since each block has its own load address, data can be loaded in widely separated memory locations even for a short load file. Thus the load file doesn't necessarily have the same number of sectors as the resulting boot sector count used by the MAKE-BOOT handler.

Finally, two more addresses are used by DOS as vectors for load file execution—the initialization address loaded to INITAD, memory location 738 ($02E2); and the run address loaded to RUNAD, memory location 736 ($02E0). The latter is executed after the file is completely loaded and the former is executed as soon as a new address is loaded to INITAD.

Generally these addresses should correspond to the boot sector run address of bytes 4 and 5 and the initialization routine starting at byte 6. Both addresses are loaded as any other data from a load file (e.g. as a two-byte data block).

HOW IT WORKS

In Part 1 of this series, we discussed three steps of adding a new handler:

1. Write the program for the handler.
2. Set up the Handler Table.
3. Make an entry in the Handler Address Table.

In step 1, the routines that comprise the handler are on lines 5000-8600.

The open routine (BOPEN, lines 5185-5480) sets the initial values of the variables used in the program and checks to make sure you still want to proceed. It also writes zeros into as many sectors as you want, starting with sector 1.

The close routine, line 7130-7495, writes the last sector to the boot disk. Then it takes the actual sector count, the run address and the initialization address, and asks you if you want to add these to the boot disk (the first nine bytes of sector 1).

The PUT BYTE routine (BPUT), line 6925-7085, receives all the data from the load file. Most subroutines in this program support the PUT BYTE handler. This routine first stores the byte from the CIO and then checks to see if it's part of the first six bytes of the load file. If so, subroutine FSTSIX checks for a load file and lets you set the sector count, load address, run address and initialization address.

Data after those first six bytes is either program data stored in a 128-byte buffer before being written to the boot disk, or load information extracted by the subroutine LDINFO. This subroutine compares the starting address of the load file with the corresponding boot sector load address and calculates the location of the next block on the boot disk. If a load file address is lower than the specified boot disk load address, an error message is issued and the CIO returns control to you.

The data in memory locations 736-739 ($02E0—$02E3) are stored in the variables RUNADR and INTADR. In the CLOSE routine, you can add these values to their respective positions in sector 1.

The handler for the GET status routine is also used as the general exit routine for all handler routines. This large program needs an internal status variable. Error codes are stored in STATS and loaded into the accumulator and Y register when the handler returns to the CIO. The CIO returns control to you when an error code greater than 127 occurs.

The GET BYTE and special functions are not implemented here and are represented by NOFNT (line 6785). This is simply an RTS which passes error code 146 back to the CIO.

The handler in lines 5035-5065 is not very complicated. Each address is represented by the address-minus-one of each routine and is in the order given in *Figure 1* in part one of this series from last month's **Antic**.

Step 3 (lines 440-630) makes an entry into the Handler Address Table, finding an empty spot in the Handler Address Table and adding the ASCII code for "B" followed by the address of the Handler Table. That's the same routine used in the NULL handler.

I/O WITHOUT CIO

The I/O subroutines for the MAKE-BOOT handler run from line 7505 to the end of the program. The first one reads and writes sectors to the boot disk. It doesn't use the resident disk handler (DSKINV) but instead uses the serial bus I/O utility vector (SIOV) and lets you write without write-verification, greatly speeding the process of writing to disk.

To use SIOV, we must fill in all the values of the Device Control Block (DCB) from memory locations 768—779 ($0300—$030B). But for this application only four bytes of data are variable. To read a sector, set the following memory locations:

Label	Memory Location	Value
Read a Sector		
DCOMND	770 ($0302)	82 ($52)
DSTATS	771 ($0303)	64 ($40)
Write a Sector		
DCOMND	770 ($0302)	87 ($57)
DSTATS	771 ($0303)	128 ($80)

The command for writing without verification is $0050, and with verification it's $0057. The only other variable is the sector number in bytes $030A and $030B (low byte, high byte) taken from the variable SECNUM. All other values are supplied by the routine DISKIO.

The second I/O subroutine in lines

7840-7930 accepts either Y or N from the keyboard buffer and loads the accumulator with either a one for Y or a zero for N. Upon returning from this subroutine, a BEQ or BNE tests for the key pressed. The only drawback to this is that the character for the key pressed is not displayed.

These two routines perform I/O consistently between Atari operating systems. SIOV is a vector that always points to the serial bus I/O utility, and the keyboard buffer is always at $02FC. To get or display a string of bytes from the keyboard, we need a different approach.

SCREEN EDITOR HANDLER

Both writing to the screen and receiving a string from the keyboard can be done via the screen editor handler. Printing to the screen is done by loading the accumulator with the ASCII value of the character to be displayed and doing a JSR to the screen editor's PUT BYTE routine. To get a byte from the screen, do a JSR to the screen editor's GET BYTE routine. Upon return, the ASCII value of the next key pressed will be in the accumulator.

For most Atari operating systems, the screen editor's PUT BYTE routine starts at $F6A4 and the GET BYTE routine starts at $F63E. Your program might use these addresses to read and write to the screen. The problem is that these locations aren't guaranteed and may be at different locations in different operating systems. We have to find these handler routines independently of the operating system.

The method for this is included in the initialization routine for the MAKEBOOT handler. Lines 690-1070 first locate the screen editor's Handler Table by searching the Handler Address Table (starting at $031A) for the E: device.

The two bytes following the ASCII E are the address of the Handler Table, in which bytes 4 and 5 are the address-minus-one of the GET BYTE routine and bytes 6 and 7 are the address-minus-one of the PUT BYTE routine. These addresses are stored in a three-byte jump instruction on lines 8295 and 8320. One is added to each address, so we're ready to do I/O to and from the screen.

You could use this program to produce your own bootable games or programs that produce a disk menu.

Instead of doing a JSR to a location in the operating system, we do a JSR to either EPUT or EGET. The program is vectored to the true address of the PUT BYTE and GET BYTE routine.

Now that we've established a legal way of using the screen editor to read and write to the screen, we can finish discussing the I/O routines.

To use the subroutine in lines 8140-8270 that displays characters, load the *low* byte of the address of the first character of the string into the accumulator and the *high* byte into the Y register. Then JSR to PRINT.

This continues to display characters until it finds one with the most significant bit set (values greater than 127). If the last character equals 128, then the cursor will remain at the end of that line of text. All values greater than 128 will make the text end with a carriage return. The only other control character is a carriage return, represented by zero. Lines 8260-8515 give examples of how this routine is used.

The routine called PNUM (lines 7950-7975) displays a two-byte integer as a base 10 number. To use it, put the low byte of the number in $00D4 (FR0) and the high byte in $00D5. Then do a JSR to PNUM. The routine uses the floating-point routines found at $D800 to $DFFF. IFP converts the integer to a floating-point number in FR0. FASC converts a floating-point number in FR0 to a string in a buffer called INBUFF at $0580. PRINTE displays the resulting string.

Finally, GETNUM inputs a user-generated number and converts it to an integer in FR0. This routine also uses the floating-point routines, but it starts with an ASCII string in INBUFF. The string is input from the keyboard by doing a JSR to EGET until a carriage return is reached. To avoid most errors, the ASCII value for a zero ($0030) is put in the first byte in the INBUFF buffer. This means that any character other than a number will return a zero.

USING THE PROGRAM

MAKEBOOT is written for OS/A+ DOS, and will not work with Atari DOS 2. Compile the source code (Listing 1, MAKEBOOT.M65) using MAC/65 or your Atari Assembler/Editor cartridge. If you have the Antic monthly disk, you will find both the source code and the executable file (MAKEBOOT.EXE) already on the disk. (This executable file will NOT run with DOS 2.)

Load the resulting file from DOS. To use the new handler, simply use the COPY command to copy the desired load file to the B: handler. You need a disk to hold the new boot sectors. (It's a good idea to use a freshly formatted disk, and always a good idea to work with backup copies of your programs, just in case.)

The first six bytes of information can be added in several ways. The easiest is to put them in your program before compiling it, as shown in the example below.

```
100 ;Start of your program
110 ;
120 *= (your load address)
130 START .BYTE 0
140 SECCNT .BYTE [LAST-START]/
    128+1
150 LOADAD .WORD START
160 RUNADR .WORD (your run
    address)
170 INITAD JMP (your program
    init)
170 ;
```

LAST is a label that is added to the end of your program.

If this is impossible or inconvenient, they can be added while the MAKEBOOT handler is running. The first opportunity is before the load file is written to the boot disk. At this point the program asks you for the sector count, load address, run address and initialization address.

If you use this method, you must leave at least six bytes between your load address and the beginning of the boot program for the boot information. If you specify an initialization address, you must leave nine bytes because the initialization address is added as a jump to the address you specify and it starts at byte 6.

Finally, after the boot sectors are written, the MAKEBOOT handler gives you the actual sector count, load file run address and load file initialization address. Then it asks you if you want to add them. If you respond with a [Y] to these prompts, the corresponding data will be added to sector 1. ▲

Listing on page 38

FIGURE 1 Boot Sector Data

Byte #	Bytes	Purpose
0	1	Flag stored at $0240
1	1	Boot sector count
2	2	Load address
4	2	Run address
6	3	Jump to initialization address

The initialization address can start at byte 6 but for the purposes of this program, a jump instruction is placed here.

NEW ARRIVALS

NEW XL/XE SOFTWARE!

DARK CHAMBERS
CROSSBOW (XE gun)
ONE ON ONE
CHOPLIFTER
AIRBALL SUMMER GAMES
CRIME BUSTERS (XE gun)
KARATEKA
AIRBALL

1020 COLOR PRINTER PLOTTER

Complete with:
- 2 Pen Sets
- 1 Roll Paper
- Power Supply & Cable

$14.98 Brand New

EXTRA PEN SETS .... $ .89 (Black) $3.98 (Color)

ATTN. DEALERS

PAC-MAN Case of 120 **$69.**

800 48K COMPUTER w/PAC-MAN **$69.95** Reconditioned

1025 PRINTER
- 80 Column
- Dot Matrix
- Friction/Tractor
- Direct Connect

$79.98 Reconditioned

1200XL 64K COMPUTER

w/PAC-MAN **$49.** Reconditioned

1027 PRINTER

- 80 Column
- Direct Connect
- Letter Quality

$79. Brand New

1050 DISK DRIVE $169.

Reconditioned

800 BOARDS

- Mother
- Power
- CPU
- ROM

YOUR CHOICE **$8.98**

8Bit I/O Cable . $4.98

850 Interface Serial, parallel $79.98 Reconditioned

810 DRIVES $129. Reconditioned

BASIC TUTOR 4 BOOKS $4.98

5.25" DISKS 20 CENTS EA.*

QTY.	PRICE
10	$4.00
100	$29.95
*1000	$200

MAJORITY ARE UNNOTCHED CONTAINING OLD SOFTWARE

NEW DOS XE .. $9.95

DOS 2.5 w/manual .. $4.98

1010 Tape drive ...... $29.
10' Joystick ext. ...... $ .99
Trackball $9.95

CARTRIDGES FOR 800, XL, XE

BASIC CARTRIDGE	$4.95	CHICKEN	$8.98	MARIO BROS.	$19.98	STAR RAIDERS II	$19.98
BASIC TUTOR (4 BOOKS)	$4.95	CLAIM JUMPER	$8.98	DONKEY KONG JR.	$19.98	DAVID'S MIDNIGHT MAGIC	$19.98
TURMOIL	$4.95	DELTA DRAWING	$8.98	JUNGLE HUNT	$19.98	ARCHON	$19.98
PAC-MAN (no box)	$4.95	HEY DIDDLE DIDDLE	$8.98	MOON PATROL	$19.98	KARATEKA	$19.98
DONKEY KONG (no box)	$4.95	SLIME (400/800)	$8.98	BATTLEZONE	$19.98	CHOPLIFTER	$19.98
GORF (400,800)	$4.95	ALPHABET ZOO	$8.98	FOOD FIGHT	$19.98	GATO	$24.98
DEMON ATTACK (400,800)	$4.95	ALF	$8.98	HARDBALL	$19.98	ACE OF ACES	$24.98
DELUXE INVADERS	$4.95	ADVENTURE CREATOR	$8.98	FIGHT NIGHT	$19.98	LODE RUNNER	$24.98
JOURNEY TO THE PLANETS	$4.95	SKY WRITER	$14.95	ONE ON ONE BASKETBALL	$19.98	BARNYARD BLASTER (XE gun)	$24.98
MATH ENCOUNTER	$7.98	FOOTBALL	$14.95	DESERT FALCON	$19.98	DARK CHAMBERS	$29.98
DANCE FANTASY	$8.98	DEFENDER	$14.95	NECROMANCER	$19.98	AIRBALL	$29.98
LOGIC LEVELS	$8.98	ROBOTRON	$19.98	RESCUE ON FRACTALUS	$19.98	SUMMER GAMES	$29.98
MEMORY MANOR	$8.98	TENNIS	$19.98	BALLBLAZER	$19.98	CROSSBOW (XE gun)	$29.98
LINKING LOGIC	$8.98	FINAL LEGACY	$19.98	BLUE MAX	$19.98	CRIME BUSTERS (XE gun)	$29.98

DISK SOFTWARE FOR 800, XL, XE

DAVID'S MIDNIGHT MAGIC	$4.98	CROSSCHECK	$4.98	SPIDERMAN	$4.98
ZORRO	$4.98	MOLECULE MAN	$4.98	HULK	$4.98
BANDITS (48K 400,800)	$4.98	CRYSTAL RAIDER	$4.98	VISICALC	$24.98
PROTECTOR II	$4.98	DESPATCH RIDER	$4.98	BOOKKEEPER W/ numeric keypad	$29.98
CLAIM JUMPER	$4.98	MISSION ASTEROID	$4.98	GET RICH	$39.98
SYNTREND	$4.98				

ONLY $4.95 **8-bit Bargains!**

Commbat
Preppie I & II
The Count
Disk 50 (50 Games)
Freaky Factory
Laser Hawk

Special buys on 8-BIT Software!

SAN JOSE COMPUTER

THE ATARI STORE

Sunrise Plaza 640 Blossom Hill Rd. San Jose, CA 95123
(408) 224-8575 • BBS (408) 224-9052

ONLY $4.95 **8-bit Bargains!**

Rocket Repairman
Adventureland
Pirate Adventure
Secret Mission
Voodoo Castle
Technicolor Dream

SHIPPING: ADD $5.00 TO ALL ORDERS. AIR AND INTERNATIONAL SHIPPING EXTRA. THAT'S IT.
WARRANTY: 90 DAY WARRANTY ON ALL ITEMS. **TAX:** CALIFORNIA RESIDENTS ADD 7% SALES TAX.
PREPAYMENT: USE VISA, MASTERCARD, MONEY ORDER, CASHIER'S CHECK OR PERSONAL CHECK.
PERSONAL CHECK MUST CLEAR PRIOR TO SHIPMENT. **C.O.D.:** CASH, CASHIER'S CHECK OR M.O. ONLY.
Prices subject to change without notice. Brand and/or product names are trademarks or registered trademarks of their respective holders.
Ad produced on an ATARI ST using Publishing Partner and printed on an ATARI SLM804 PostScript compatible laser printer.

DISK BONUS

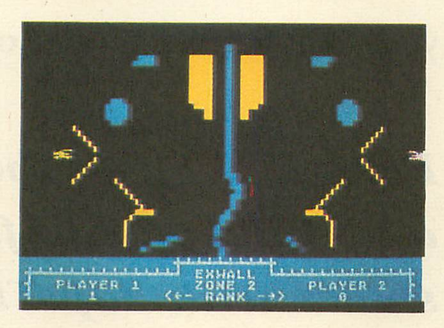
Exwall

Futuristic tank battle for two players.

Program by Andy Barton

Battle it out for construction space, in this month's Super Disk Bonus. Exwall is a hard-hitting futuristic two player tank game written in assembly language. The program works on 8-bit computers with at least 48K memory, disk drive and two joysticks.

Exwall was written in fast-moving machine language by Andy Barton, who has been contributing to **Antic** since 1984. His programs include *Spelling Flashcards* (February 1989) and the ever-popular *TYPO II*. His Machine Language Stringer appears elsewhere in this issue.



THE SCENARIO

Across eight battle zones, the robot war continues without end. Powerful robots battle it out for building space, over a terrain marked with stone fortress walls and blue rivers and lakes. Massive mobile factories, the robots move about, building and repairing the fortress walls, even as they try to destroy each other.

Your robot carries a cannon that fires explosive shells—but so does your opponent's. In this high-tech slugfest, you must destroy your opponent's fortresses and robot factory without being destroyed yourself. Your ranking reflects your success.

GETTING STARTED

Don't try to run Exwall directly from the Antic Monthly Disk. Instead, copy EXWALL.EXE to another disk that includes the DOS.SYS file. Use DOS command E to rename EXWALL.EXE to AUTORUN.SYS. Turn off your computer and remove all cartridges. Place the disk in Drive 1. Hold down the [OPTION] key if you're using an XL or XE. Turn on the computer, and Exwall will load and run automatically.

Plug in two joysticks, one for each player. Use your joystick to move your robot factory up and down, forward, backward and diagonally. You can't move through walls, over water, or into the farthest quarter of your opponent's territory.

To build a wall, move the robot factory to the desired place. Then move back 1 or 2 spaces and stop. A two-layer wall will be built in front of the machine. Construction will stop when the area in front of the machine and within the crane's reach is completed—or when you move the factory.

Press the joystick button once to fire your cannon. Press it a second time to explode the shell. Press and release the fire button quickly, or the shell will explode before you want it to. You can use unexploded shells to knock narrow channels in your own defenses, and then fire through the channels as you hide in the relative safety of your fortress.

You can't move over water but you can build walls at the water's edge. When these walls are destroyed the debris will fill in the water and allow you to move on.

A shell exploding next to a wall will penetrate deeply but not widely. Conversely, if the shell explodes before it reaches a wall it will destroy a wider area but with less penetration.

Crashing your robot into your opponent's will destroy both machines —and possibly get you out of a tight situation. But it won't help your ranking. On the other hand, daring players can significantly enhance their scores by moving in close and firing repeatedly into their opponent's exploding robot.

Your July 1989 Antic Disk—featuring the Exwall Super Bonus game as well as every type-in program from this issue—will be shipped to you within 24 hours after receiving your order. Just phone Toll-Free to the Antic Disk Desk at **(800) 234-7001**. The monthly disk is only $5.95 (plus $2 for shipping and handling) on your Visa or MasterCard. Or mail a $5.95 check (plus $2 shipping and handling) to Antic Disk Desk, 544 Second Street, San Francisco, CA 94107. ▲

TapeTime LabelMaker

Printing your VCR log. By Gary Coppola

Print VCR labels showing titles, speeds, and times of your favorite movies and shows with TapeTime Labelmaker. A simple modification of* Antic's *TapeTime program (January 1989), TapeTime Labelmaker will make managing your VCR collection easier than ever. Works on 8-bit Atari computers with at least 48K memory, disk drive and an Epson-compatible printer.

TapeTime Labelmaker is a label printing utility that will allow you to print the title, speed and time of your favorite videotaped movies or television programs on any Beta or VHS label. I developed the label maker as an add-on to Paul Shannon's *TapeTime* (**Antic**, January 1989) which determines the amount of time remaining on partially recorded VCR tapes.

When **Antic** challenged readers to design a patch for an add-on labeler, I thought this should be no problem! A dozen or so lines of code and presto, a labelmaker. Well, after rummaging through all of my VCR tapes and checking out their labels, I realized that in nearly every instance, no two brands provided the same area and location to print in. In fact, even different tape grades of the same brand had different labels. So much for an easy patch.

What evolved after several approaches to the problem was a user-friendly program that automatically selects the print size on the label according to the number of titles you have entered. This allows you to have up to six titles on Beta format labels and up to seven titles on VHS labels. The positioning of the printing area is accomplished with only one user input. It's that easy!

GETTING STARTED

If you don't already have *TapeTime* simply type in Listing 2, VCRLABEL.BAS, and be sure to SAVE a copy to disk before you run it.

If you already have TapeTime, the patch can be typed and merged with the original listing. Type in Listing 1, TAPEMOD.LST, check it with TYPO II and LIST it to disk. LOAD the original TapeTime program and then ENTER "D:TAPEMOD.LST". Don't forget to SAVE the merged file back to disk as "D:VCRLABEL.BAS".

Several lines of TapeTime have been rewritten in order to accommodate the labelmaking patch. You'll find the complete TapeTime Labelmaker on this month's Antic Disk, ready to RUN without any modifications or merging required.

MAKING A LABEL

When you RUN the modified version of TapeTime the first choice you will have to make is either Do Calculation or Make a Label. If you select Do Calculation you will proceed ex-

actly as in the original TapeTime program.

If you select Make a Label you will next be prompted to choose either Beta or VHS format. After typing your choice, a list of the more popular VCR tape brands will appear onscreen with a column of numbers under the heading of CMAX. This is a suggested value for the maximum number of characters your title should contain in order for it to fit within the allowable space on that brand's label.

After a short time, a scale with numbers ranging from 5 to 35 appears in the message area with room to enter your title right below it. This scale helps you easily determine the number of characters in your title as you are typing it. After entering your first title you will be asked for its speed and time.

This sequence repeats, allowing you to enter up to a maximum of six titles for Beta and seven titles for VHS. If you have less than the maximum, just press [RETURN] when prompted for the next title and you will enter the printing mode.

At this point, insert a label into the printer with the brand name or logo facing to the right. Align the label's left edge with column 0 (on many printers the column scale is located on the tear bar).

HELPFUL HINTS

Now, turn on the printer and, at the prompt, enter the margin setting. A list of VCR tape brands along with some suggested settings will appear. That's about it—press any key and your label will be printed.

If your brand of tape is not among those in the list, make a trial label on paper to insure that your margin setting and title length are correct. An easy way to find the appropriate margin setting is to hold your label next to the printers' column scale with its left edge on column 0. Then read the column number where you want your printing to begin. This is the margin setting.

Once you determine satisfactory margin settings and title lengths for your particular brand of label, you can replace some of the brands on the list with your own brands and values. This way, you won't forget those numbers the next time you use the program. The lists are located in Lines 3015-3070 in Listing 1.

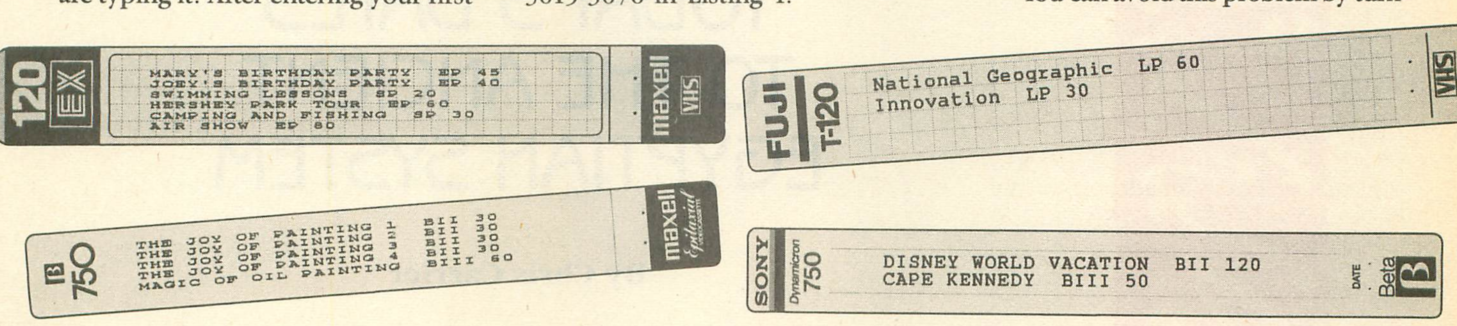
The size of type on the label is controlled by the number of titles you enter. In Beta format, if you have three titles or less, they will be printed in normal-size type. For more than three titles, the print size will be one-half that of normal. In VHS format, four titles or less will be in normal-size type, otherwise you will get the smaller type. If you expect the label to print in the smaller type, I recommend that you use all capital letters in your title for better readability.

After entering your title and speed you are prompted for the program length (in minutes). This was done to keep the same format as in the calculation portion of the program. In the labelmaking portion, however, this is not critical. If you would rather type 2hr instead of 120 (minutes), go right ahead but remember you only have a maximum of three characters. The printer control codes used in this program are Epson codes. If you own a printer that is not Epson compatible you can replace the codes in Lines 3200-3245 with your own printer codes. Each code has a REM statement following it so you know exactly what each code represents. Just look in your printer manual for the proper codes.

When the program enters the printing mode and you input the margin setting, you will notice that the printer advances the label approximately five-eighths of an inch, as a result of the printer receiving the control codes. If you don't take this into account you may have to reposition your label at this point.

You can avoid this problem by turning off the printer's line feed DIP switch when you first place your label in the printer. (My interface has a line feed switch on it also, which makes this process easier.) After entering the margin setting the printer will now receive the codes without moving the label. Turn the line feed switch back on and press any key to print the label.

If you need more room than is allotted on your brand name label, blank VHS pin-feed labels (sorry, Beta owners) are available in a variety of colors from SlideScribe, 7141 Shady Oak Road, Minneapolis, MN 55344. ▲



Gary Coppola of Budd Lake, New Jersey is a senior research chemist for a major pharmaceutical company. He has written two books and has over 75 scientific publications and patents.

Listing on page 29



EGYPT CALENDAR

CONVERTS TODAY'S DATES TO THE ANCIENT EGYPTIAN SYSTEM

By Chris Carrier

Feel like a Pharoah! Convert any date from the Julian or Gregorian calendars back to the calendars used in ancient Egypt. This BASIC program works on Atari 8-bit computers with at least 48K memory.

Your Atari can make you an Egyptologist. Egypt Calendar takes any date in the Julian or the modern Gregorian calendars, and gives the equivalent dates in both of the civil calendars used in ancient Egypt, one of which survives even to this day.

Just enter the date, and a screen full of information about that day and year appears. Not only does the program give the original Egyptian calendar date, but the year is given as determined by the reigns of several ancient kings, or by the Alexandrian, Augustan and Coptic calendars. The program can even tell you the day of the week on which that date fell, or the Julian Day number, useful for astronomers.

GETTING STARTED

Type in Listing 1, EGYPT.BAS, check it with TYPO.II, and SAVE a



copy to disk before you RUN it.

The program will then ask you to input a date. If the date is between 1582-1923, you will be prompted to specify whether the date uses the Julian or Gregorian calendar. The Gregorian calendar began in 1582 A.D., but the last nation to switch from Julian to Gregorian (Greece) did not do so until 1923.

Then press [RETURN] to see how your date translates. At the top of your screen the date you entered will be displayed, followed by the Julian Day Number. Used by chronologists and astronomers, this number simply tells you how many days have passed since January 1, 4713 B.C. (JD #0).

The original Egyptian calendar date follows, with the year as determined by eight different eras. At the very bottom of the screen is the date of the Sothic Rising, the astrological phenomenon used by the ancient Egyptians to track the actual, as opposed to calendar, year.

During the first couple of millenia of the calendar's existence, the months had no names, but were simply referred to as the first, second, third. . .month of their season.

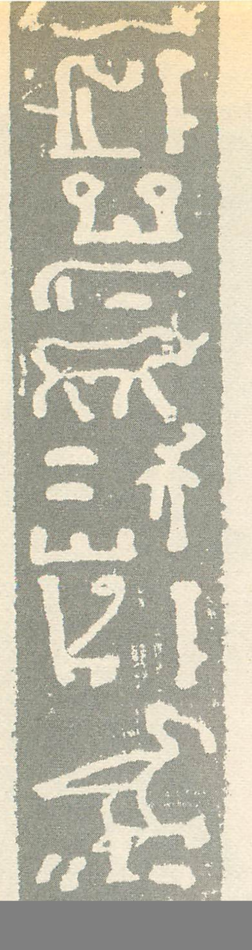
EGYPTIAN DATES

The ancient Egyptians, among their other accomplishments, were probably the first people in the history of the world to discover the number of days in a year down to the nearest integer.

The original version of the Egyptian calendar had a week of 10 days, a month of 3 weeks or 30 days, a season defined as 4 months or 120 days, three seasons equaling 360 days, which were followed by five unnamed epagomenal, or "outside the calendar," days to total 365 days in a year.

The year began with the season of Akhit (Flood, as in Nile River), followed by Perit (Winter) and Shemu (Summer). Egypt Calendar gives this date thus: Shemu 2-14 for the 14th day of the second month in the summer (Shemu) season. The five epagomenal days are treated by the program as a five-day, fifth month of Shemu. Because a tradition developed that any work done on the epagomenal days was unlucky, the ancient Egyptians ended their year with a five-day festival.

The great advantage of the Egyptian calendar is that it was easy to use—it survived in daily use for more than 3,000 years. Astronomers and historians used the calendar for convenience of chronology as late as the 16th century. France tried a version of it shortly after the French Revolution, in the late 18th century. This calendar has even drawn praise from



a 20th century astronomer (Neugebeuer, "A History of Ancient Mathematical Astronomy") for being the "most sensible of all calendars used by mankind", with its easy-to-use 10-day weeks, 30-day months, and 365-day years.

CORRECTIONS NEEDED

The great disadvantage of Egyptian calendars is that the number of days in the year is not 365, but rather 365.2422 days in a tropical year (the cycle of the seasons) or 365.2564 days in a sidereal year (one orbit of Earth with respect to the stars).

Therefore, the Egyptian calendar ran about 1 day fast every 4 years, so if an annual event occurred on, say, Akhit 1-1 in a given year, it would occur on Akhit 1-2 after four years, Akhit 1-3 after eight years, during the second month of Akhit after about

length of the sidereal year has become known, and the true Sothic Cycle is 1422 Old Egyptian years long. The next time a Sothic Rising coincides with Akhit 1-1 will be on August 27, 2985 A.D.

Also of note is the fact that the Sothic Rising occurs in the Julian calendar about 1 day later every 150 years, and in our Gregorian calendar, 1 day later every 72 years. This is because of the precession of the equinoxes, the same phenomenon which makes Polaris the North Star in our lifetimes and Thuban the North Star during the building of the Great Pyramid.

Although any estimates as to exactly when the Egyptian calendar began to function are only educated guesses, Sothic Risings on Akhit 1-1 occurred in 1282-85 B.C., 2706-09 B.C., and 4130-33 B.C. The pr

Egyptian government or its subjects. (In fact, the World Almanac 1989 gives in "Chronological Cycles" the opening of the Nabonassarian Year 2738 on April 26, 1989—a fact that can be checked using Egypt Calendar.)

When Augustus Caesar conquered Egypt, shortly after the Roman Empire adopted a 365.25 day year, he decreed that a 366th day be added to the Egyptian calendar in every fourth year, thereby setting up a new 365.25 day calendar with an Augustan Era (from Augustus' accession) and an Alexandrian Era (from Rome's conquest of Egypt).

After Augustus' death, however, the new calendar went ignored for almost 300 years, until the Coptic Christian Church in Egypt decided to start using it, with an era beginning with the reign of the then-current Roman emperor, Diolectian. Since Diolectian is primarily remembered for persecution of and atrocities against Christians, the Coptics refer to their era, which they use even today, as the "Era of Martyrs." The year 1706 of the Coptic Era of Martyrs begins on Sept. 11, 1989.

Egypt Calendar gives the date in the revised 365.25-day Egyptian calendar, and as names for the months were in use by the time of the reform, these month names are used instead of the season and month number. The year in the New Egyptian calendar begins on August 29 or 30 in the Julian calendar, and between 1900 and 2099 A.D. in the Gregorian calendar, on September 11 or 12. The Alexandrian, Augustan, and Coptic eras are all given by the program.

This program was inspired by the chapter on the Egyptian calendar in O. L. Harvey's "Calendar Conversions by way of the Julian Day Number."

LIST OF VARIABLES

CAL$—the calendar in which the date is being inputted, either Julian or Gregorian. Anyone entering a date between 1582-1923 will be prompted for the calendar they want.

ENSEASON$—The season of the year, in the original Egyptian calendar.

EMO—The number of the month in the season in the original Egyptian calendar.

EBM—The day of the month in the original Egyptian calendar.

D$—The day of the (seven day) week.

M$—The month of the year in the Julian or Gregorian calendar.

DATE—The day of the month, Julian or Gregorian.

IYEAR—The number of the year, Julian or Gregorian.

ERA$—A.D. or B.C.

JDAY—Julian Day Number, used by chronologists and astronomers, and is a linear count without end. JD #0 was January 1, 4713 B.C.—in the late 20th century A.D., the number is between 2440000 and 2450000.

JSC—Julian Sothic Cycle of Censorinus. 1461 original Egyptian years long.

TSC—True Sothic Cycle. 1424 original Egyptian years long.

EY—Number of the original Egyptian year in the current Julian Sothic Cycle.

TEY—Number of the original Egyptian year in the current True Sothic Cycle.

SENWORSET3—Number of original Egyptian years since the coronation of Senworset III.

AMENHOTEP1—Number of original Egyptian years since the coronation of Amentohep I.

NABONASSAR—Number of original Egyptian years since the coronation of Nabonassar.

CY—Year of the Coptic Era of Martyrs, expressed in the New Egyptian calendar with a 366th day every 4th year.

ENMONTH$—The month of the Egyptian year, expressed in the 365.25 day New Egyptian calendar.

CBM—The day of the month in the New Egyptian calendar.

ALEXANDRIAN—The day of the month in the New Egyptian calendar, counting from Augustus' conquest of Alexandria and his attempt to institute a 365.25 day calendar in Egypt.

AUGUSTAN—The day of the month in the New Egyptian calendar, counting from year 1 of Augustus' reign as Emperor of Rome.

SRMONTH$—The Julian or Gregorian month in which Sirius makes its first appearance in the early morning (the Sothic Rising) just before dawn, after having been invisible for a month or so because it was above the horizon during daylight hours only.

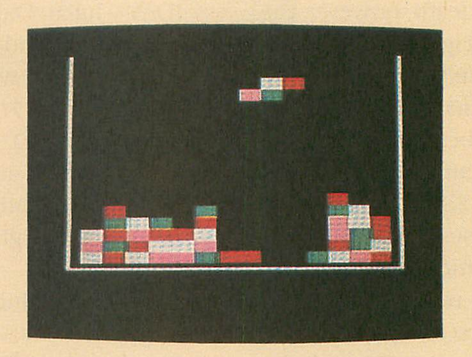
SR—The day of the month on which the Sothic Rising occurs. ▲

Chris Carrier lives in Sacramento, California. His interests include astronomy, chronology and games. His articles have appeared in USA Today and the Barrow Sun, the northernmost newspaper in North America. This is his first appearance in Antic.

Listing on page 28

Red Squares

Challenge for your mind and your reflexes. **By Marc Abramowitz**



Tired of shoot-'em-ups and mazes? This colorful strategy game, adapted from the popular Russian import, challenges both your mind and reflexes. This BASIC program works on 8-bit Atari computers with at least 48K and disk drive.

Are you sick and tired of shoot-'em-up games? Do the words "Space Invaders" drive you crazy? Well, if you're tired of senseless violence, ringing ears and joystick-cramped hands, try Red Squares, a challenging strategy game that challenges your mind and reflexes.

Red Squares is based on an imaginative Russian computer game. A popular American version has been released for several makes of personal computers including the Atari ST—but not for the 8-bit Atari.

But now this oversight is remedied. Red Squares lets you play this colorful and exciting new game on your 8-bit Atari. The game features 50 speeds and other options for added challenge.

GETTING STARTED

Type in Listing 1, SQUARES.BAS, and check it with TYPO II. Be sure to SAVE a copy before you RUN it.

If you have trouble typing the special characters in lines 100 and 8010, don't type them in. Instead, type Listing 2, check it with TYPO II and SAVE a copy. When you RUN Listing 2, it creates these hard-to-type lines and stores them in a file called LINES.LST.

To merge the two programs, LOAD "D:SQUARES.BAS" and then ENTER "D:LINES.LST." Remember to SAVE the completed program before you RUN it.

Various shapes, all made of four squares, fall into a pit. The object is to move and rotate the shapes, guiding them into position at the bottom of the pit. Whenever the shapes fill a complete line across the bottom of the pit, that line disintegrates, and the pieces above move down. The game is lost when the pieces are piled up all the way to the top of the pit.

You get a certain number of points for completing a line, depending on how close it is to the top of the pit. The closer the line is to the top of the pit, the more points you get. You also score one point for each shape that falls into the pit. Even if you can't complete many lines, you should try to survive as long as possible.

When you first RUN Red Squares, press [START] or the joystick button

to reach the options screen. Move the joystick up and down or press [SELECT] to change the speed, which ranges from 1 to 50. As you play, the speed increases gradually, so don't start at too high a speed.

Moving the joystick right and left or pressing [OPTION] lets you add extra challenge to the game, by starting the game with one to ten layers of randomly-placed pieces in the bottom of the pit.

FALLING SQUARES

Once you've selected speed and height, press [START] or the joystick button to begin the game. The pit will appear, and pieces will start falling from above, one by one.

Move pieces right or left using the joystick, or by pressing [J] and [L]. To rotate a piece 90 degrees, press the joystick trigger, or [K]. Pieces may be rotated several times—if the shape changes with such rotation. The square doesn't rotate, since the shape would remain the same.

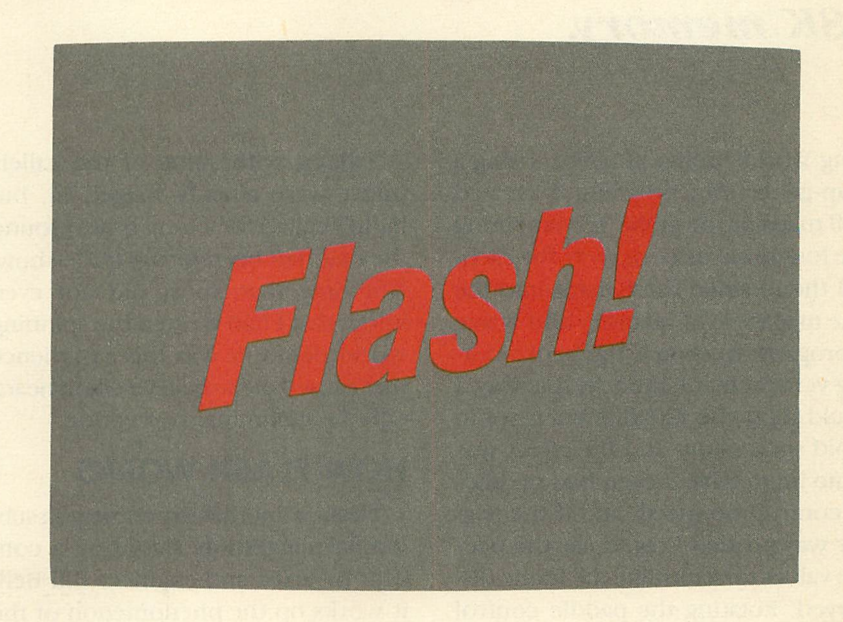
Try to leave as few gaps as possible when landing pieces. The more lines you complete, the more room you'll have for future pieces—and the more you manage to get into the pit, the higher your score.

These rules are simple, but play can be challenging. With a little practice, you'll know which pieces will fit which spaces best—but the piece you need most may take its time about arriving. As the pieces fall faster and faster you'll find yourself struggling to place each piece just so. It takes intense concentration and some sharp thinking to truly master the falling squares.

Marc Abramowitz is a Freshman at Port Richmond High School in Staten Island, New York. Besides computers, his interests include playing the guitar and fantasy role-playing games—particularly Teenage Ninja Mutant Turtles and Advanced Dungeons and Dragons. This is his first appearance in Antic.

Listing on page 33

Light show with a hidden message.



By Ernie Negus

Many people who use computers are highly analytical thinkers. There is nothing wrong with that, but new ideas that are difficult to prove seem a little off-the-wall to them. I myself was very skeptical about subliminal perception. I simply did not believe the claims that some people made about it. After reading a book on the subject, I decided to try out my own experiment with subliminal messages.

Flash! is one of my experiments, a short machine-language program that lets you enter a message, which is then flashed on the screen in a series of dots.

GETTING STARTED

Type in Listing 1, FLASH.BAS, check it with TYPO II and SAVE a copy before you RUN it. When RUN, FLASH.BAS creates a machine language program called FLASH.EXE. Antic Disk owners will find FLASH.EXE on the monthly disk.

Don't try to run Flash! from the Antic Monthly Disk. Copy FLASH.EXE to another disk that has the DOS.SYS file on it. Then rename FLASH.EXE to AUTORUN.SYS.

Turn your Atari off and place your Flash! disk in drive 1. Now, turn on your Atari and Flash! will automatically load and start.

CONVINCING EXPERIENCE

My first experiment with subliminal perception convinced me that it worked. At the time, about two years ago, I upgraded my 130XE to 1Mb of memory. I was also running a 24-hour bulletin board.

To show off the power of a 130XE with one megabyte, I wrote a "Spin-

Give yourself a subliminal boost—or test the power of subliminal messages on your family and friends with Flash! This BASIC program creates a machine language program that works on 8-bit Atari computers with at least 48K memory.

ning World" demo program. Using a map-generating program, I created 120 maps of the globe, incrementing the longitude three degrees for each.

I then loaded these maps into the one megabyte of memory and wrote a program to switch the banks during vertical blank time. In this way, I could show the 120 different maps in rapid succession, and the effect was quite impressive. I even had paddles to control the speed, and if the trigger was pressed I could use the paddle values to determine the frame displayed. Rotating the paddle control yielded a very weird effect, as the globe would eerily rotate along with it.

I was working at a computer store, demonstrating my program at various trade shows. It was quite a popular display. As an experiment, I added another frame that would flash for a few milliseconds just before the first frame when the globe was spinning at full speed.

At the time, my BBS was very inactive. I was lucky to get three or four calls a day. So naturally my subliminal suggestion was, "Call BEE-CATS BBS!" written in huge letters on the extra frame.

The effect was a slight flickering of the screen when the demo was running. The flicker was barely perceptible and far too fast to read no matter how hard you looked.

The demo with its subliminal message ran at a trade show for three days. The night of the first day I had fifty calls on my BBS. The next day, I had over 200!

Talking with some of the callers (most were already logged in, but hadn't called for a long time) I found they all had been to the trade show. Not only that, most did not even know that I had written the spinning world demo! It was this experience that caused me to believe wholeheartedly in subliminal perception.

HOW FLASH WORKS

Flash! is another experiment in subliminal suggestion. Based on a concept by artist and engineer Bill Bell, it works on the phenomenon of the eye's perception of light and the way the subliminal mind can interpret the patterns from the eye.

After you run the program you will be prompted for a phrase. Enter a single word or a short phrase. For the best results, keep the phrase as short as possible, with not more than four words. After that, the screen clears and a column of eight boxes will begin to flash rapidly on the screen.

Most people who see this will just think it's a pretty display. If they try to concentrate on the flickering boxes to get any meaning they will probably think even less of it. But when they look away and start thinking about something else, the words you typed may suddenly appear in their imagination.

To understand how the program works, visualize one of those electric news signs, like the one in New York's Times Square, where words travel across a panel. The sign contains several columns of bulbs that turn on and off in a special sequence to produce the moving message effect.

Now visualize just looking at one of the columns of lights. After a certain amount of time, the whole message gets scrolled through that one column. But it looks like just a bunch of flickering lights.

Your subliminal mind is able to perceive the message being scrolled through that single column of lights, but your conscious mind cannot comprehend it as a message. It is only when you look away, when you may finally get the message coming through.

This makes Flash! a great program for developing your subconscious mind. For example, if you want to lose weight, use the phrase, "I will be thin" on Flash!. If you are trying to develop a more positive attitude try "I am happy." Try to avoid negative words such as "not" or "no" or "won't" — these words tend to confuse the subconscious mind and will often produce the opposite of the desired effect.

To exit Flash! while it is displaying your phrase, just press any key. To slow down the message, press the [SELECT] key and release it several times. To speed it up, press the [OPTION] key several times. The audible click that occurs happens at the end of the phrase, just before the message is recycled through.

Ernie Negus is a longtime contributor to Antic. Currently he lives in Bothell, Washington, working as a technician on state-of-the-art ultrasound equipment.

Listing on page 36

SOFTWARE LIBRARY

TYPING SPECIAL ATARI CHARACTERS

The Atari Special Characters and the keys you must type in order to get them are shown below:

For [CONTROL] key combination, *hold down* [CONTROL] while pressing the next key. For inverse [CONTROL] [A] through [CONTROL] [Z], press the [◪] key—or [⅄] on the 400/800—then *release* it before pressing the next key. (Press [◪] or [⅄] again to turn off inverse.) For [ESC] key combinations, press [ESC] and then *release* it before pressing the next key.

Carefully study the chart above and pay close attention to differences between lookalike characters such as the slash key's [/] and the [CONTROL] [F] symbol [◩].

NORMAL VIDEO

FOR THIS	TYPE THIS
	CTRL ,
	CTRL A
	CTRL B
	CTRL C
	CTRL D
	CTRL E
	CTRL F
	CTRL G
	CTRL H
	CTRL I
	CTRL J
	CTRL K
	CTRL L
	CTRL M
	CTRL N
	CTRL O
	CTRL P
	CTRL Q
	CTRL R

FOR THIS	TYPE THIS
	CTRL S
	CTRL T
	CTRL U
	CTRL V
	CTRL W
	CTRL X
	CTRL Y
	CTRL Z
	ESC ESC
	ESC CTRL −
	ESC CTRL =
	ESC CTRL +
	ESC CTRL *
	CTRL .
	CTRL ;
	SHIFT =
	ESC SHIFT CLEAR
	ESC DELETE
	ESC TAB

INVERSE VIDEO

FOR THIS	TYPE THIS
	ESC SHIFT DELETE
	ESC SHIFT INSERT
	ESC CTRL TAB
	ESC SHIFT TAB
	⅄ CTRL .
	⅄ CTRL ;
	⅄ SHIFT =
	ESC CTRL 2
	ESC CTRL DELETE
	ESC CTRL INSERT

TYPO II AUTOMATIC PROOFREADER

TYPO II automatically proofreads **Antic**'s type-in BASIC listings. Type in the listing below and SAVE a copy to disk or cassette. Now type GOTO 32000. At the prompt, type in a single program line **without the two-letter TYPO II code at the beginning**. Then press [RETURN].

Your line will reappear at the bottom of the screen. If the TYPO II code does not match the code in the magazine, then you've mistyped your line.

To call back a previously typed line, type [*], then the line number, then [RETURN]. When the completed line appears, press [RETURN] again. This is how TYPO II proofreads itself.

To LIST your program, press [BREAK] and type LIST. To return to TYPO II, type GOTO 32000. To remove TYPO II from your program, type LIST "D:FILENAME",0,31999, then [RETURN], then NEW, then ENTER "D:FILENAME", then [RETURN]. Now you can SAVE or LIST your program to disk or cassette.

Don't type the TYPO II Codes!

```
WB 32000 REM TYPO II BY ANDY BARTON
VM 32010 REM VER. 1.0 FOR ANTIC MAGAZINE
HS 32020 CLR :DIM LINE$(120):CLOSE #2:CLO
   SE #3
BN 32030 OPEN #2,4,0,"E":OPEN #3,5,0,"E"
YC 32040 ? "↖":POSITION 11,1:? "TYPO II"
EM 32050 TRAP 32040:POSITION 2,3:? "Type
   in a program line"
HS 32060 POSITION 1,4:? " ":INPUT #2;LINE
   $:IF LINE$="" THEN POSITION 2,4:LIST B
   :GOTO 32060
XH 32070 IF LINE$(1,1)="*" THEN B=VAL(LIN
   E$(2,LEN(LINE$))):POSITION 2,4:LIST B:
   GOTO 32060
TH 32080 POSITION 2,10:? "CONT"
MF 32090 B=VAL(LINE$):POSITION 1,3:? " ";
NY 32100 POKE 842,13:STOP
CN 32110 POKE 842,12
ET 32120 ? "↖":POSITION 11,1:? " TYPO II 
   ":POSITION 2,15:LIST B
CE 32130 C=0:ANS=C
QR 32140 POSITION 2,16:INPUT #3;LINE$:IF
   LINE$="" THEN ? "LINE ";B;" DELETED":G
   OTO 32050
VV 32150 FOR D=1 TO LEN(LINE$):C=C+1:ANS=
   ANS+(C*ASC(LINE$(D,D))):NEXT D
WJ 32160 CODE=INT(ANS/676)
JW 32170 CODE=ANS-(CODE*676)
EH 32180 HCODE=INT(CODE/26)
BH 32190 LCODE=CODE-(HCODE*26)+65
HB 32200 HCODE=HCODE+65
IE 32210 POSITION 0,16:? CHR$(HCODE);CHR$
   (LCODE)
VG 32220 POSITION 2,13:? "If CODE does no
   t match press  RETURN  and edit line a
   bove.":GOTO 32050
```

Egypt Calendar

Article on page 20

Article on page 20

LISTING 1

Don't type the TYPO II Codes!

```
MF 5 REM EGYPTIAN CALENDAR
ID 10 REM BY CHRIS CARRIER
PJ 15 REM (C)1989, ANTIC PUBLISHING INC.
DU 20 DIM D$(9),M$(9),ERA$(5),CAL$(18),ENMONTH$(15),ENSEASON$(7),SRMONTH$(9)
QD 25 POKE 710,14:POKE 712,14:POKE 709,14:? "[clear]":? :? :?
ZE 35 ? "This will give you the date in both Egyptian calendars, and the ";
BG 37 ? "heliacal rising of Sirius."
VG 38 ? :? :? "Type in a YEAR, MONTH, DATE"
HW 39 ? "(If date is BC, type a '-' before the year.)":?
HH 40 ? "YEAR ";:GOSUB 410:INPUT YEAR:? "MONTH ";:INPUT MONTH:? "DATE ";:INPUT DATE
FH 41 IF YEAR<1582 THEN CAL$="J":GO TO 45
CI 42 IF YEAR>1923 THEN CAL$="G":GO TO 45
UX 43 ? :? " Gregorian or Julian Calendar? ";
MO 44 INPUT CAL$
SI 45 IYEAR=YEAR
CD 48 IF IYEAR<0 THEN IYEAR=0-IYEAR:BC=1
GB 49 IF BC=0 THEN ERA$="A.D."
FS 50 IF BC=1 THEN ERA$="B.C."
IH 60 IF YEAR<0 THEN YEAR=YEAR+1
PW 70 JYEAR=YEAR+4712
EJ 80 JDAY=0
EP 90 JDAY=JDAY+(JYEAR*365)
VS 95 REM *** ADD LEAP YEARS
GR 100 LDAYS=1178+INT(YEAR/4)
XK 110 JDAY=JDAY+LDAYS
JY 130 JDAY=JDAY+2
SD 135 IF CAL$="J" THEN 160
UN 140 JDAY=JDAY-INT(YEAR/100)
WH 150 JDAY=JDAY+INT(YEAR/400)
JK 160 IF MONTH=2 THEN JDAY=JDAY+31
DN 161 REM LEAP YEAR
CM 163 IF YEAR/4=INT(YEAR/4) THEN X=1:DATE=DATE+1
ZQ 164 IF CAL$="J" THEN 168
MK 165 IF YEAR/400=INT(YEAR/400) THEN X=1:DATE=DATE+1
LL 168 IF X=1 THEN DATE=DATE-1
ZN 169 IF X=1 AND MONTH>2 THEN JDAY=JDAY+1
WG 170 IF MONTH=3 THEN JDAY=JDAY+59
QO 180 IF MONTH=4 THEN JDAY=JDAY+90
TA 190 IF MONTH=5 THEN JDAY=JDAY+120
XW 200 IF MONTH=6 THEN JDAY=JDAY+151
CE 210 IF MONTH=7 THEN JDAY=JDAY+181
WQ 220 IF MONTH=8 THEN JDAY=JDAY+212
CF 230 IF MONTH=9 THEN JDAY=JDAY+243
YG 240 IF MONTH=10 THEN JDAY=JDAY+273
SO 250 IF MONTH=11 THEN JDAY=JDAY+304
XA 260 IF MONTH=12 THEN JDAY=JDAY+334
HJ 270 JDAY=JDAY+DATE
EH 271 IF CAL$="J" THEN JDAY=JDAY-2
BU 272 IF X=1 THEN JDAY=JDAY-1
VZ 275 IF CAL$="J" THEN 290
VC 276 IF YEAR/400=INT(YEAR/400) THEN 280
WI 277 IF YEAR/100=INT(YEAR/100) AND MONTH<3 THEN JDAY=JDAY+1
DW 280 IF YEAR/400=INT(YEAR/400) THEN JDAY=JDAY-1:DATE=DATE-1
CQ 290 DOW=JDAY
YY 300 IF DOW>700000 THEN DOW=DOW-700000
HT 305 IF DOW<1 THEN DOW=DOW+700000
JQ 310 IF DOW>70000 THEN DOW=DOW-70000
BS 320 IF DOW>7000 THEN DOW=DOW-7000
BE 330 IF DOW>700 THEN DOW=DOW-700
IA 340 IF DOW>70 THEN DOW=DOW-70
WT 345 IF DOW>7 THEN DOW=DOW-7
SO 346 IF DOW>7 THEN GOTO 300
UE 348 IF DOW<1 THEN GOTO 305
DQ 350 RESTORE 1100:FOR CJ=1 TO DOW:READ D$:NEXT CJ:GOTO 500
MX 400 FOR CJ=2 TO 14:POKE 709,CJ:NEXT CJ:RETURN
TS 410 FOR CJ=14 TO 2 STEP -1:POKE 709,CJ:NEXT CJ:RETURN
LC 500 RESTORE 1000:FOR CJ=1 TO MONTH:READ M$:NEXT CJ
MT 1000 DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY,AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
VO 1100 DATA TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,SUNDAY,MONDAY
SW 1200 DATA THOUT,PAAPE,HATOR,KIAHK,TOBE,MSHIR,PARMHAT,PARMUTE,PASHONS,PAONE,EPEP,MESORE,EPAGOMENAL
JC 2000 REM ORIG. EGYPTIAN CALENDAR
GA 2001 EBM=JDAY-171867
UQ 2002 GOSUB 400:? "[clear]":POSITION 12,12:? "CALCULATING...";:GOSUB 410
ES 2010 IF EBM<0 THEN EBM=EBM+533265:EY=EY-1461
IP 2015 IF EY<0 THEN JSC=JSC-1:EY=EY+1461
KO 2020 IF EBM>533265 THEN EBM=EBM-533265:EY=EY+1461
VB 2030 IF EBM>36500 THEN EBM=EBM-36500:EY=EY+100
LT 2040 IF EBM>3650 THEN EBM=EBM-3650:EY=EY+10
TB 2050 IF EBM>365 THEN EBM=EBM-365:EY=EY+1
BL 2060 IF EBM>365 THEN 2010
QD 2061 TEY=EY-111
ZB 2065 IF EY>1461 THEN JSC=JSC+1:EY=EY-1461
GM 2066 IF EY>1461 THEN 2065
BK 2067 IF TEY>1424 THEN TSC=TSC+1:TEY=TEY-1424
XS 2068 IF TEY>1424 THEN 2067
AT 2070 IF EBM<1 THEN 2010
HZ 2100 REM
FU 2105 ESEASON=1
VR 2110 IF EBM>120 THEN EBM=EBM-120:ESEASON=ESEASON+1
UX 2120 IF EBM>120 THEN 2110
MC 2125 REM 5 EPAGOMENAL DAYS MADE
RR 2126 REM A 5TH MONTH OF SUMMER
QM 2127 IF ESEASON=4 THEN ESEASON=3:EBM=EBM+120
KR 2130 IF ESEASON=1 THEN ENSEASON$=" AKHIT "
MI 2131 IF ESEASON=2 THEN ENSEASON$=" PERIT "
KC 2132 IF ESEASON=3 THEN ENSEASON$=" SHEMU "
MZ 2200 REM MONTH AND DAY
WO 2205 EMO=1
JR 2210 IF EBM>30 THEN EBM=EBM-30:EMO=EMO+1
ZK 2220 IF EBM>30 THEN 2210
QH 2300 REM COMPUTE ERAS
OL 2310 SENWORSET3=((JSC-1)*1461)-904+EY
VL 2320 AMENTOHEP1=((JSC-1)*1461)-1238+EY
FO 2330 NABONASSAR=((JSC-1)*1461)-2036+EY
UC 3000 REM 365.25 DAY CALENDAR
TG 3100 CBM=JDAY-1824664
PX 3110 IF CBM<0 THEN CBM=CBM+146100:CY=CY-400
ZK 3120 IF CBM>146100 THEN CBM=CBM-146100:CY=CY+400
FI 3130 IF CBM>14610 THEN CBM=CBM-14610:CY=CY+40
BW 3140 IF CBM>1461 THEN CBM=CBM-1461:CY=CY+4
BU 3150 IF CBM>1461 THEN 3120
BD 3160 IF CBM<0 THEN 3110
SN 3200 FOR CJ=1 TO 3:IF CBM>365 THEN CBM=CBM-365:CY=CY+1
KP 3201 NEXT CJ
FV 3210 IF CBM>30 THEN CM=CM+1:CBM=CBM-30:IF CBM>30 THEN 3210
RV 3230 CM=CM+1
PX 3300 REM NAMES OF MONTHS
MS 3310 RESTORE 1200:FOR CJ=1 TO CM:READ ENMONTH$:NEXT CJ
BO 3700 REM OTHER ERAS
HM 3710 ALEXANDRIAN=CY+308
VC 3720 AUGUSTAN=CY+312
CW 4000 REM SOTHIC RISING
```

```
ZK 4010 SR=199.365967+(YEAR*(549.5/86400)
   )
VJ 4015 IF CAL$="G" THEN SR=SR+INT(YEAR/1
   00)-2
CR 4016 IF CAL$="G" THEN SR=SR-INT(YEAR/4
   00)
LQ 4100 IF SR>365.25 THEN SR=SR-365.25
EA 4101 IF SR>365.25 THEN 4100
KX 4102 IF SR<0 THEN SR=SR+365.25
HV 4103 IF SR<0 THEN 4102
KG 4110 SRMONTH$="JANUARY"
HL 4120 IF SR>31 THEN SRMONTH$="FEBRUARY"
ZJ 4130 IF SR>59.25 THEN SRMONTH$="MARCH"
PF 4140 IF SR>90.25 THEN SRMONTH$="APRIL"
SC 4150 IF SR>120.25 THEN SRMONTH$="MAY"
XN 4160 IF SR>151.25 THEN SRMONTH$="JUNE"
YV 4170 IF SR>181.25 THEN SRMONTH$="JULY"
NZ 4180 IF SR>212.25 THEN SRMONTH$="AUGUS
   T"
GY 4190 IF SR>243.25 THEN SRMONTH$="SEPTE
   MBER"
WI 4200 IF SR>273.25 THEN SRMONTH$="OCTOB
   ER"
PN 4210 IF SR>304.25 THEN SRMONTH$="NOVEM
   BER"
QH 4220 IF SR>334.25 THEN SRMONTH$="DECEM
   BER"
NK 4320 IF SRMONTH$="FEBRUARY" THEN SR=SR
   -31
QJ 4330 IF SRMONTH$="MARCH" THEN SR=SR-59
   .25
UK 4340 IF SRMONTH$="APRIL" THEN SR=SR-90
   .25
GN 4350 IF SRMONTH$="MAY" THEN SR=SR-120.
   25
OQ 4360 IF SRMONTH$="JUNE" THEN SR=SR-151
   .25
II 4370 IF SRMONTH$="JULY" THEN SR=SR-181
   .25
RP 4380 IF SRMONTH$="AUGUST" THEN SR=SR-2
   12.25
XF 4390 IF SRMONTH$="SEPTEMBER" THEN SR=S
   R-243.25
SZ 4400 IF SRMONTH$="OCTOBER" THEN SR=SR-
   273.25
DG 4410 IF SRMONTH$="NOVEMBER" THEN SR=SR
   -304.25
DT 4420 IF SRMONTH$="DECEMBER" THEN SR=SR
   -334.25
XP 4900 SR=INT(SR*1)
ER 4910 SR=SR+1
EY 9000 IF CAL$="J" THEN CAL$="JULIAN CAL
   ENDAR"
RJ 9010 IF CAL$="G" THEN CAL$="GREGORIAN 
   CALENDAR"
IU 9800 REM
OM 9810 GOSUB 400:? "K":? D$;", ";M$;" ";
   DATE;", ";IYEAR;" ";ERA$;" "
AK 9815 ? CAL$,"JD#";JDAY
AH 9900 ? "Original Calendar Date: ";ENSE
   ASON$;EMO;"-";EBM
AL 9910 ? :? "Various Eras:"
BX 9915 ? "Year #";EY;" of Julian Sothic 
   Cycle #";JSC
JV 9916 ? "Year #";TEY;" of True Sothic C
   ycle #";TSC
BT 9920 ? "Year of Senworset III: ";SENWO
   RSET3
YX 9925 ? "Year of Amentohep I:   ";AMENT
   OHEP1
MR 9930 ? "Year of Nabonassar:    ";NABON
   ASSAR
BU 9950 ? :? :? :? :?
RF 9960 ? "Coptic Calendar: ";CY;" ";ENMO
   NTH$;" ";CBM
PX 9962 IF CM=13 THEN 9965
OX 9965 ? "Alexandrian Year ";ALEXANDRIAN
HQ 9970 ? "Augustan Year      ";AUGUSTAN
PX 9990 ? :? "Sothic Rising this year: ";
   SRMONTH$;" ";SR:GOSUB 410
HY 9999 END
```

TapeTime LabelMaker

Article on page 18

LISTING 1

Don't type the TYPO II Codes!

```
LH 32 REM TAPE TIME LABELMAKER
AR 34 REM BY GARY COPPOLA
RH 36 REM (c)1989, ANTIC PUBLISHING
QS 320 REM
IK 370 POSITION 12,12:? "12800"
UJ 381 DIM TITLE1$(40),TITLE2$(40),TITLE3
   $(40),TITLE4$(40),TITLE5$(40),TITLE6$(
   40),TITLE7$(40)
JO 382 DIM SPEED1$(4),SPEED2$(4),SPEED3$(
   4),SPEED4$(4),SPEED5$(4),SPEED6$(4),SP
   EED7$(4)
ML 383 DIM MIN1$(3),MIN2$(3),MIN3$(3),MIN
   4$(3),MIN5$(3),MIN6$(3),MIN7$(3)
EI 385 LINE=13:POSITION 2,16:? "Do You Wa
   nt To:  1)Do Calculation":POSITION 19,
   17:? "2)Make a Label"
MD 386 TRAP 1920:POSITION 2,18:? "Your Ch
   oice";:INPUT CH:GOSUB 1750
ZM 387 IF CH<1 OR CH>2 THEN 381
OK 388 IF CH=1 THEN GOSUB 1750:GOSUB 1830
   :GOTO 390
HK 389 IF CH=2 THEN POSITION 12,12:? "LAB
   EL":GOSUB 1710:GOSUB 1750:GOTO 2170
JK 1430 IF X=1 THEN GOSUB 1710:GOSUB 1750
   :GOTO 380
NP 1750 FOR CS=16 TO 18:POSITION 0,CS:? "
    ":NEXT CS:RETURN
OA 1960 IF ERROR=8 THEN ON LINE GOTO 390,
   480,510,610,2060,2060,880,910,1010,108
   0,1380,1470,381,2170,2850
CC 2170 LINE=14:POSITION 2,16:? "Tape For
   mat:  1)BETA    2)VHS"
KT 2180 TRAP 1920:POSITION 2,17:? "Your C
   hoice";:INPUT MODE:GOSUB 1750
HW 2190 IF MODE<1 OR MODE>2 THEN 2170
FT 2200 IF MODE=1 THEN POSITION 12,12:? "
   BETA■":GOSUB 1750:GOSUB 3000
WF 2210 IF MODE=2 THEN POSITION 12,12:? "
   ■VHS■":GOSUB 1750:GOSUB 3000:GOTO 2310
PF 2220 REM BETA
KI 2260 GOSUB 3015:GOSUB 3020
YE 2270 FOR WAIT=1 TO 500:NEXT WAIT:GOSUB
    1750:POSITION 2,16:? "You can enter a
    maximum of 6 titles"
YM 2280 ? "    Press ANY KEY to continue"
QE 2290 CLOSE #2:OPEN #2,4,0,"K:":GET #2,
   K:CLOSE #2:GOTO 2400
RB 2300 REM VHS
LI 2310 GOSUB 3025:GOSUB 3030
BF 2330 FOR WAIT=1 TO 500:NEXT WAIT:GOSUB
    1750:POSITION 2,16:? "You can enter a
    maximum of 7 titles"
SQ 2340 GOTO 2280
WE 2400 GOSUB 1750:POSITION 2,17:? "Enter
    Your FIRST Title":FOR WAIT=1 TO 90:NE
   XT WAIT
SJ 2410 TRAP 2400:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE1$:IF TITLE1$="" THEN GOT
   O 2790
FB 2420 TNUM=1:GOSUB 1750:GOSUB 3120
BH 2440 INPUT SPEED1$
EP 2460 GOSUB 3150:INPUT MIN1$
GN 2470 GOSUB 1750:POSITION 2,17:? "Enter
    Your SECOND Title":FOR WAIT=1 TO 90:N
   EXT WAIT
```

```
BC 2480 TRAP 2470:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE2$:IF TITLE2$="" THEN GOT
   O 2790
GH 2490 TNUM=2:GOSUB 1750:GOSUB 3120
BO 2500 INPUT SPEED2$
FC 2510 GOSUB 3150:INPUT MIN2$
WR 2520 GOSUB 1750:POSITION 2,17:? "Enter
   Your THIRD Title":FOR WAIT=1 TO 90:NE
   XT WAIT
CZ 2530 TRAP 2520:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE3$:IF TITLE3$="" THEN GOT
   O 2790
GF 2540 TNUM=3:GOSUB 1750:GOSUB 3120
CU 2550 INPUT SPEED3$
GR 2560 GOSUB 3150:INPUT MIN3$
GW 2570 GOSUB 1750:POSITION 2,17:? "Enter
   Your FOURTH Title":FOR WAIT=1 TO 90:N
   EXT WAIT
KM 2580 TRAP 2570:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE4$:IF TITLE4$="" THEN GOT
   O 2790
HF 2590 TNUM=4:GOSUB 1750:GOSUB 3120
CY 2600 INPUT SPEED4$
HE 2610 GOSUB 3150:INPUT MIN4$
GC 2620 GOSUB 1750:POSITION 2,17:? "Enter
   Your FIFTH Title":FOR WAIT=1 TO 90:NE
   XT WAIT
MJ 2630 TRAP 2620:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE5$:IF TITLE5$="" THEN GOT
   O 2790
HD 2640 TNUM=5:GOSUB 1750:GOSUB 3120
EE 2650 INPUT SPEED5$
IT 2660 GOSUB 3150:INPUT MIN5$
JS 2670 GOSUB 1750:POSITION 2,17:? "Enter
   Your SIXTH Title":FOR WAIT=1 TO 90:NE
   XT WAIT
TW 2680 TRAP 2670:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE6$:IF TITLE6$="" THEN GOT
   O 2790
ID 2690 TNUM=6:GOSUB 1750:GOSUB 3120
EI 2700 INPUT SPEED6$
JG 2710 GOSUB 3150:INPUT MIN6$
CW 2720 IF MODE=2 THEN GOSUB 1750:POSITIO
   N 2,17:? "Enter Your SEVENTH Title":FO
   R WAIT=1 TO 90:NEXT WAIT
VL 2730 IF MODE=2 THEN TRAP 2720:GOSUB 30
   80:POSITION 1,18:INPUT TITLE7$:IF TITL
   E7$="" THEN GOTO 2790
ND 2740 IF MODE=2 THEN TNUM=7:GOSUB 1750:
   GOSUB 3120:INPUT SPEED7$
AQ 2750 IF MODE=2 THEN GOSUB 3150:INPUT M
   IN7$
EE 2790 GOSUB 1750
JU 2795 POSITION 2,16:? "Position LABEL i
   n the printer,then"
ZD 2800 POSITION 10,17:? " TURN ON PRINTE
   R ":FOR J=15 TO 0 STEP -1:SOUND 0,120,
   10,J:NEXT J
EB 2810 FOR WAIT=1 TO 300:NEXT WAIT
KM 2830 IF MODE=1 AND TNUM<=3 THEN LS=12
TP 2835 IF MODE=1 AND TNUM>3 THEN LS=6
MD 2840 IF MODE=2 AND TNUM<=4 THEN LS=12
VF 2845 IF MODE=2 AND TNUM>4 THEN LS=6
NP 2850 LINE=15:GOSUB 1750
RX 2854 IF MODE=1 THEN GOSUB 3050
TW 2856 IF MODE=2 THEN GOSUB 3060
TL 2860 TRAP 1920:POSITION 9,16:? " SET T
   HE LEFT MARGIN ":? "Refer to list for
   suggested setting"
JV 2870 ? "MARGIN=";:INPUT MAR
VF 2880 IF MAR<1 OR MAR>40 THEN 2850
QA 2890 GOTO 3200
FO 3000 POSITION 4,16:? "Refer to list fo
   r the suggested":? " maximum number o
   f characters"
HK 3010 ? " (CMAX) allowable for your ti
   tle":RETURN
MM 3015 POKE 82,20:POSITION 20,2:? " Bran
   d":? " --------------":? " SONY":? :?
   " SCOTCH"
DK 3017 ? :? " FUJI":? :? " MAXELL":POKE
   82,2:RETURN
IA 3020 POKE 82,32:POSITION 31,2:? "CMAX"
   :? :? "31":? :? "20":? :? "31":? :? "3
   0":POKE 82,2:RETURN
IO 3025 POKE 82,20:POSITION 20,2:? " Bran
   d":? " ---------------":? " SCOTCH":? "
    SCOTCH EG":? " MAXELL EX"
HP 3027 ? " MAXELL HGX":? " FUJI":? " MEM
   OREX":? " POLAROID":? " PANASONIC":POK
   E 82,2:RETURN
MF 3030 POKE 82,32:POSITION 31,2:? "CMAX"
   :? :? "35":? "27":? "30":? "27":? "35"
   :? "31"
XS 3040 ? "28":? "30":POKE 82,2:RETURN
RD 3050 POKE 82,31:POSITION 29,2:? "MARGI
   N":? :? " 8 ":? :? " 10 ":? :? " 9 ":?
   :? " 8 ":POKE 82,2:RETURN
DY 3060 POKE 82,32:POSITION 30,2:? "MARGI
   N":? :? " 3 ":? " 3 ":? " 11 ":? " 14 ":
   ? " 10 "
EG 3070 ? " 8 ":? " 18 ":? " 12 ":POKE 82,2
   :RETURN
SW 3080 POSITION 2,17:? "----------------
   --------------------"
RB 3090 POSITION 2,16:? "   5    10   15
   20   25   30   35":RETURN
GP 3100 POSITION 2,16:? "Enter Speed (BII
   or BIII):";:RETURN
DX 3110 POSITION 2,16:? "Enter Speed (SP,
   LP or EP):";:RETURN
LY 3120 IF MODE=1 THEN GOSUB 3100
NS 3130 IF MODE=2 THEN GOSUB 3110
AP 3140 RETURN
OW 3150 POSITION 2,17:? "Program Length (
   in minutes)";:RETURN
NG 3199 REM PRINTER CONTROL CODES
FZ 3200 LPRINT CHR$(27);CHR$(64):REM INIT
   IALIZE PRINTER
TW 3210 LPRINT CHR$(27);CHR$(110):REM NLQ
   PICA
GP 3220 LPRINT CHR$(27);CHR$(65);CHR$(LS)
   :REM LS/72 LINE SPACING
JU 3230 LPRINT CHR$(27);CHR$(77);CHR$(MAR
   ):REM LEFT MARGIN
CV 3240 IF MODE=1 AND TNUM>3 THEN LPRINT
   CHR$(27);CHR$(83);CHR$(1):REM SUBSCRIP
   T FONT
FC 3245 IF MODE=2 AND TNUM>4 THEN LPRINT
   CHR$(27);CHR$(83);CHR$(1):REM SUBSCRIP
   T FONT
CB 3250 GOSUB 1750:POSITION 6,17:? "Press
   ANY KEY to print LABEL"
LM 3260 CLOSE #2:OPEN #2,4,0,"K:":GET #2,
   K:CLOSE #2
XR 3270 REM PRINT LABEL
YB 3280 LPRINT TITLE1$;"  ";SPEED1$;" ";M
   IN1$;" "
BQ 3290 LPRINT TITLE2$;"  ";SPEED2$;" ";M
   IN2$;" "
ED 3300 LPRINT TITLE3$;"  ";SPEED3$;" ";M
   IN3$;" "
PK 3310 IF TNUM>3 THEN LPRINT TITLE4$;"
   ";SPEED4$;" ";MIN4$;" "
US 3320 IF TNUM>3 THEN LPRINT TITLE5$;"
   ";SPEED5$;" ";MIN5$;" "
AA 3330 IF TNUM>3 THEN LPRINT TITLE6$;"
   ";SPEED6$;" ";MIN6$;" "
GV 3340 IF TNUM>3 AND MODE=2 THEN LPRINT
   TITLE7$;"  ";SPEED7$;" ";MIN7$;" "
ZM 3350 GOSUB 1710:GOSUB 1750:GOTO 380
```

LISTING 2

```
YY 10 REM VCR TAPE TIME
ES 20 REM BY PAUL SHANNON
QH 30 REM (c)1988, ANTIC PUBLISHING
LH 32 REM TAPE TIME LABELMAKER
AR 34 REM BY GARY COPPOLA
RH 36 REM (c)1989, ANTIC PUBLISHING
BA 40 GRAPHICS 0:POKE 752,1:POKE 710,0:DI
   M ERR$(6),HOLD$(6)
UN 50 ? "
   "
GZ 60 ? "
   "
FJ 70 ? "
   "
MT 80 ? "   APE  IMED
   "
EF 90 ? "
   "
RS 100 ? "     A  VCR
   "
GU 110 ? "  UTILITY BY
   "
VX 120 ? "  PAUL SHANNON
   "
CT 130 ? "  (c)1988 Antic
   "
MI 140 ? "
```

```
   ███▯▯ "
WG 150 ? "▯▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▯▯██████████████
   ███▯▯ "
LO 160 ? "▯▯▯▯▯▯▯▯▯▯▯▯▬▬▬▬▬▬▬▬▯▯██████████████
   ███▯▯ "
BU 170 ? "▯          ▯          ▯▯▬▬▬▬▬▬▬▬▬▬▬▬▬▬
   ▬▬▬▬▯ "
ZH 180 ? "▯ ▯▯▯▯▯  ▬▬▬▬▬▬▯▯ █████         █
   ██ ▯ "
QK 190 ? "▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
   ▬▬▬▬▬ "
DN 200 ? "▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
   ▬▬▬▬▬"
RI 210 POSITION 2,19:? "████████████████████
   ██████████████████"
UL 220 POSITION 2,20:? " ▯
                    ▯ "
VE 230 POSITION 2,21:? " ▯
                    ▯ "
QT 240 POSITION 2,22:? " ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
   ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ "
JE 250 FOR J=1 TO 10:FOR L=2 TO 11
LA 260 POSITION 20,L:? "▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
   :POSITION 20,L:? "████████████████"
HM 280 NEXT L
GD 300 NEXT J
QS 320 REM
PI 330 FOR Z=1 TO 5:POSITION 12,12:? "12:
   00":FOR WAIT=1 TO 100:NEXT WAIT
ZE 340 POSITION 12,12:? "█████":FOR WAIT=
   1 TO 100:NEXT WAIT
MT 360 NEXT Z
IK 370 POSITION 12,12:? "12:00"
MF 380 CLR :DIM ERR$(6),HOLD$(6)
UJ 381 DIM TITLE1$(40),TITLE2$(40),TITLE3
   $(40),TITLE4$(40),TITLE5$(40),TITLE6$(
   40),TITLE7$(40)
JO 382 DIM SPEED1$(4),SPEED2$(4),SPEED3$(
   4),SPEED4$(4),SPEED5$(4),SPEED6$(4),SP
   EED7$(4)
ML 383 DIM MIN1$(3),MIN2$(3),MIN3$(3),MIN
   4$(3),MIN5$(3),MIN6$(3),MIN7$(3)
EI 385 LINE=13:POSITION 2,16:? "Do You Wa
   nt To:  1)Do Calculation":POSITION 19,
   17:? "2)Make a Label"
MD 386 TRAP 1920:POSITION 2,18:? "Your Ch
   oice";:INPUT CH:GOSUB 1750
ZM 387 IF CH<1 OR CH>2 THEN 381
OK 388 IF CH=1 THEN GOSUB 1750:GOSUB 1830
   :GOTO 390
HK 389 IF CH=2 THEN POSITION 12,12:? "LAB
   EL":GOSUB 1710:GOSUB 1750:GOTO 2170
GC 390 POSITION 2,16:? "Do You Want: 1)BE
   TA 2)VHS"
LY 400 LINE=1:TRAP 1920
US 410 POSITION 27,16:INPUT A:GOSUB 1750:
   GOSUB 1830
QE 420 IF A<1 OR A>2 THEN 390
DV 430 IF A=1 THEN POSITION 12,12:? "BETA
   █"
RH 440 IF A=2 THEN POSITION 12,12:? "█VHS
   █"
ZC 450 IF A=1 THEN GOSUB 1830
AK 460 IF A=2 THEN 880
LX 470 REM BETA
MY 480 LINE=2:TRAP 1920
GM 490 POSITION 2,16:? "Is Your BETA Tape
   : 1)L-750 2)L-830":POSITION 36,16:INPU
   T TAPETYPE:GOSUB 1750
ZD 500 IF TAPETYPE<1 OR TAPETYPE>2 THEN 4
   80
MB 510 POSITION 2,16:? "Enter The Minutes
    BETA II"
MX 520 LINE=3:TRAP 1920
FE 530 POSITION 27,16:INPUT MINBII:GOSUB
   1750
NF 540 IF MINBII=0 THEN GOSUB 1710:GOTO 5
   80
HC 550 IF MINBII>900 THEN GOSUB 2080:GOTO
    510
MJ 560 TOTMINBII=TOTMINBII+MINBII
NX 570 GOTO 510
FV 580 IF TAPETYPE=1 THEN POSITION 23,2:?
    "BETA L-750":POSITION 22,3:? "▬▬▬▬▬▬▬
   ▬▬▬▬▬▬"
EQ 590 IF TAPETYPE=2 THEN POSITION 23,2:?
    "BETA L-830":POSITION 22,3:? "▬▬▬▬▬▬▬
   ▬▬▬▬▬▬"
RA 600 POSITION 22,5:? "Total Minutes":PO
   SITION 21,6:? "BETA II██";TOTMINBII
NG 610 LINE=4:TRAP 1920
PL 620 POSITION 2,16:? "Enter The Minutes
    BETA III"
RT 630 POSITION 28,16:INPUT MINBIII:GOSUB
    1750
CA 640 IF MINBIII=0 THEN 670
NO 645 IF MINBIII>900 THEN GOSUB 2080:GOT
   O 620
UK 650 TOTMINBIII=TOTMINBIII+MINBIII
OR 660 GOTO 620
VQ 670 POSITION 22,8:? "Total Minutes":PO
   SITION 21,9:? "BETA III█";TOTMINBIII
YG 680 FOR PAUSE=1 TO 200:NEXT PAUSE
JZ 690 DIM TAPE$(11)
SD 700 IF TAPETYPE=1 THEN TBII=180:TBIII=
   270:MET=222:TAPE$="█BETA L-750"
KC 710 IF TAPETYPE=2 THEN TBII=200:TBIII=
   300:MET=246:TAPE$="█BETA L-830"
DE 740 BIIMETER=(TOTMINBII/TBII)*MET
EK 750 BIIIMETER=(TOTMINBIII/TBIII)*MET
DS 760 TOTALMETER=BIIMETER+BIIIMETER
ON 770 TIMELEFTBII=((MET-TOTALMETER)/MET)
   *TBII:GOSUB 1710
SV 780 RO=INT(100*TIMELEFTBII+0.5)/100
IY 790 POSITION 22,2:? TAPE$:POSITION 21,
   3:? "▬▬▬▬▬▬▬▬▬▬▬▬▬"
DX 800 POSITION 21,5:? RO;"█Minutes":GOSU
   B 1790
CG 810 POSITION 21,6:? "BETA II Left"
ML 820 TIMELEFTBIII=((MET-TOTALMETER)/MET
   )*TBIII
ZL 830 ROIII=INT(100*TIMELEFTBIII+0.5)/10
   0
PD 840 POSITION 21,8:? ROIII;"█Minutes":G
   OSUB 1790
DR 850 POSITION 21,9:? "BETA III Left":GO
   SUB 1750
OY 860 GOTO 1380
PS 870 REM VHS
PA 880 LINE=7:TRAP 1920
KM 890 POSITION 2,16:? "Is Your VHS Tape:
    1)T-120 2)T-160":POSITION 35,16:INPUT
    VHSTYPE:GOSUB 1750
YA 900 IF VHSTYPE<1 OR VHSTYPE>2 THEN 880
QI 910 POSITION 2,16:? "Enter The Minutes
    SP"
OZ 920 LINE=8:TRAP 1920
UI 930 POSITION 22,16:INPUT MINSP:GOSUB 1
   750
HK 940 IF MINSP=0 THEN GOSUB 1710:GOTO 98
   0
XT 950 IF MINSP>900 THEN GOSUB 2080:GOTO
   910
FD 960 TOTMINSP=TOTMINSP+MINSP
PP 970 GOTO 910
RW 980 IF VHSTYPE=1 THEN POSITION 23,2:?
   "VHS T-120":POSITION 21,3:? "▬▬▬▬▬▬▬▬▬
   ▬▬▬▬▬"
ZU 990 IF VHSTYPE=2 THEN POSITION 23,2:?
   "VHS T-160":POSITION 21,3:? "▬▬▬▬▬▬▬▬▬
   ▬▬▬▬▬"
OZ 1000 POSITION 21,4:? "Total Minutes":P
   OSITION 21,5:? "SP Speed██";TOTMINSP
KV 1010 LINE=9:TRAP 1920
UL 1020 POSITION 2,16:? "Enter The Minute
   s LP"
LP 1030 POSITION 22,16:INPUT MINLP:GOSUB
   1750
HP 1040 IF MINLP=0 THEN 1070
FT 1045 IF MINLP>900 THEN GOSUB 2080:GOTO
    1020
NN 1050 TOTMINLP=TOTMINLP+MINLP
OG 1060 GOTO 1020
PE 1070 POSITION 21,7:? "Total Minutes":P
   OSITION 21,8:? "LP Speed██";TOTMINLP
BR 1080 LINE=10:TRAP 1920
AW 1090 POSITION 2,16:? "Enter The Minute
   s EP(ELP)"
GI 1100 POSITION 27,16:INPUT MINEP:GOSUB
   1750
CF 1110 IF MINEP=0 THEN 1140
NO 1115 IF MINEP>900 THEN GOSUB 2080:GOTO
    1090
XC 1120 TOTMINEP=TOTMINEP+MINEP
RM 1130 GOTO 1090
KC 1140 POSITION 21,10:? "Total Minutes":
   POSITION 21,11:? "EP Speed██";TOTMINEP
CN 1150 FOR PAUSE=1 TO 200:NEXT PAUSE
AW 1160 DIM VHSTAPE$(11)
```

```
YO 1170 IF VHSTYPE=1 THEN TSP=120:TLP=240
   :TEP=360:MET=246:VHSTAPE$=" VHS T-120"
SZ 1190 IF VHSTYPE=2 THEN TSP=160:TLP=320
   :TEP=480:MET=327:VHSTAPE$=" VHS T-160"
YJ 1210 SPMETER=(TOTMINSP/TSP)*MET
KM 1220 LPMETER=(TOTMINLP/TLP)*MET
WP 1230 EPMETER=(TOTMINEP/TEP)*MET
AF 1240 TOTALMETERS=SPMETER+LPMETER+EPMET
   ER
SI 1250 TIMELEFTSP=((MET-TOTALMETERS)/MET
   )*TSP:GOSUB 1710
JL 1260 ROSP=INT(100*TIMELEFTSP+0.5)/100
SS 1270 POSITION 22,2:? VHSTAPE$:POSITION
    21,3:? "              "
SF 1280 POSITION 21,4:? ROSP;" Minutes"
QW 1290 POSITION 21,5:? "SP Speed Left"
RJ 1300 TIMELEFTLP=((MET-TOTALMETERS)/MET
   )*TLP
ZN 1310 ROLP=INT(100*TIMELEFTLP+0.5)/100
NF 1320 POSITION 21,7:? ROLP;" Minutes"
MD 1330 POSITION 21,8:? "LP Speed Left"
CT 1340 TIMELEFTEP=((MET-TOTALMETERS)/MET
   )*TEP
QO 1350 ROEP=INT(100*TIMELEFTEP+0.5)/100
NI 1360 POSITION 21,10:? ROEP;" Minutes":
   GOSUB 1810
PF 1370 POSITION 21,11:? "EP Speed Left":
   GOSUB 1750
CJ 1380 LINE=11:TRAP 1920
AZ 1390 POSITION 2,16:? "Want To 1)CONTIN
   UE 2)PRINT"
XV 1400 POSITION 28,16:INPUT X
KR 1410 IF X<1 OR X>2 THEN GOSUB 1750:GOT
   O 1380
JK 1430 IF X=1 THEN GOSUB 1710:GOSUB 1750
   :GOTO 380
BB 1460 GOSUB 2060
CU 1470 LINE=12:TRAP 1920
VO 1480 REM PRINT
SO 1490 POSITION 2,17:? "Input Tape Numbe
   r"
OM 1500 INPUT I
VK 1510 LPRINT "# ";I
XQ 1520 IF A=1 THEN 1610
XN 1530 IF VHSTYPE=1 THEN LPRINT "VHS T-1
   20"
EH 1540 IF VHSTYPE=2 THEN LPRINT "VHS T-1
   60"
CZ 1550 LPRINT "----------"
JT 1560 IF X<>2 THEN 1590
XT 1570 LPRINT ROSP;"  Minutes SP Speed R
   emaining":LPRINT ROLP;"  Minutes LP Sp
   eed Remaining"
UK 1580 LPRINT ROEP;"  Minutes EP Speed R
   emaining"
BM 1590 GOSUB 2060
WX 1600 LPRINT :GOTO 1660
TX 1610 IF TAPETYPE=1 THEN LPRINT "BETA L
   -750"
TC 1620 IF TAPETYPE=2 THEN LPRINT "BETA L
   -830"
RY 1630 LPRINT "----------"
RK 1640 LPRINT RO;"  Minutes BETA II Rema
   ining"
FY 1650 LPRINT ROIII;"  Minutes BETA III
   Remaining":GOSUB 2060
CU 1660 LPRINT :GOSUB 1830
DV 1670 GOSUB 1750
SQ 1680 GOTO 380
GA 1690 END
CN 1700 REM CLEAR SCREEN
CE 1710 FOR C=2 TO 11
QY 1720 POSITION 20,C:? "
   "
CY 1730 NEXT C
AZ 1740 RETURN
NP 1750 FOR CS=16 TO 18:POSITION 0,CS:? "

   ":NEXT CS:RETURN
BL 1780 RETURN
SA 1790 TOTMINBII=0:TOTMINBIII=0
AP 1800 RETURN
AE 1810 TOTMINSP=0:TOTMINLP=0:TOTMINEP=0
AV 1820 RETURN
EF 1830 POSITION 20,2:? "                "
AV 1840 POSITION 20,3:? " Press RETURN  "
   :POSITION 20,4:? "  After Each  "
LX 1850 POSITION 20,5:? "     Entry
     "
XQ 1860 POSITION 20,6:? "
     "
IC 1870 POSITION 20,7:? " Enter 0 When
     ":POSITION 20,8:? " Done Entering
     "
NR 1875 POSITION 20,9:? " For Each Speed
     "
JZ 1880 POSITION 20,10:? "
   "
KV 1890 POSITION 20,11:? "
   "
AR 1900 RETURN
NS 1910 REM ERROR TRAP
XT 1920 ERROR=PEEK(195)
DP 1930 GOSUB 1750
DZ 1940 IF ERROR=8 THEN POSITION 10,16:?
   "ENTER NUMBERS ONLY!":FOR WAIT=1 TO 15
   0:NEXT WAIT
DV 1950 GOSUB 1750
OA 1960 IF ERROR=8 THEN ON LINE GOTO 390,
   480,510,610,2060,2060,880,910,1010,108
   0,1380,1470,381,2170,2850
LF 2060 LPRINT "------------------------
   ---------"
AV 2070 RETURN
BG 2080 POKE 709,0:POKE 710,12:POKE 712,6
   6
CS 2090 ERR$="ERROR ":TRAP 40000
NA 2100 FOR X0=1 TO 25:P0=INT(1.5+6*((X0/
   6)-INT(X0/6))):HOLD$=ERR$(P0,6)
HN 2110 IF P0<>1 THEN HOLD$(6-P0+2)=ERR$(
   1,P0-1)
PA 2120 POSITION 12^1,12:? HOLD$(1,5):NEX
   T X0
PW 2130 POKE 712,0:POKE 709,202:POKE 710,
   0
QG 2140 IF A=1 THEN POSITION 12,12:? "BET
   A "
DJ 2150 IF A=2 THEN POSITION 12,12:? " VH
   S "
AU 2160 RETURN
CC 2170 LINE=14:POSITION 2,16:? "Tape For
   mat:  1)BETA   2)VHS"
KT 2180 TRAP 1920:POSITION 2,17:? "Your C
   hoice";:INPUT MODE:GOSUB 1750
HW 2190 IF MODE<1 OR MODE>2 THEN 2170
FT 2200 IF MODE=1 THEN POSITION 12,12:? "
   BETA ":GOSUB 1750:GOSUB 3000
WF 2210 IF MODE=2 THEN POSITION 12,12:? "
    VHS ":GOSUB 1750:GOSUB 3000:GOTO 2310
PF 2220 REM BETA
KI 2260 GOSUB 3015:GOSUB 3020
YE 2270 FOR WAIT=1 TO 500:NEXT WAIT:GOSUB
    1750:POSITION 2,16:? "You can enter a
    maximum of 6 titles"
YM 2280 ? "     Press ANY KEY to continue"
QE 2290 CLOSE #2:OPEN #2,4,0,"K:":GET #2,
   K:CLOSE #2:GOTO 2400
RB 2300 REM VHS
LI 2310 GOSUB 3025:GOSUB 3030
BF 2330 FOR WAIT=1 TO 500:NEXT WAIT:GOSUB
    1750:POSITION 2,16:? "You can enter a
    maximum of 7 titles"
SQ 2340 GOTO 2280
WE 2400 GOSUB 1750:POSITION 2,17:? "Enter
    Your FIRST Title":FOR WAIT=1 TO 90:NE
   XT WAIT
SJ 2410 TRAP 2400:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE1$:IF TITLE1$="" THEN GOT
   O 2790
FB 2420 TNUM=1:GOSUB 1750:GOSUB 3120
BH 2440 INPUT SPEED1$
EP 2460 GOSUB 3150:INPUT MIN1$
GN 2470 GOSUB 1750:POSITION 2,17:? "Enter
    Your SECOND Title":FOR WAIT=1 TO 90:N
   EXT WAIT
BC 2480 TRAP 2470:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE2$:IF TITLE2$="" THEN GOT
   O 2790
GH 2490 TNUM=2:GOSUB 1750:GOSUB 3120
BO 2500 INPUT SPEED2$
FC 2510 GOSUB 3150:INPUT MIN2$
WR 2520 GOSUB 1750:POSITION 2,17:? "Enter
    Your THIRD Title":FOR WAIT=1 TO 90:NE
   XT WAIT
CZ 2530 TRAP 2520:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE3$:IF TITLE3$="" THEN GOT
   O 2790
GF 2540 TNUM=3:GOSUB 1750:GOSUB 3120
```

```
CU 2550 INPUT SPEED3$
GR 2560 GOSUB 3150:INPUT MIN3$
GW 2570 GOSUB 1750:POSITION 2,17:? "Enter
    Your FOURTH Title":FOR WAIT=1 TO 90:N
   EXT WAIT
KM 2580 TRAP 2570:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE4$:IF TITLE4$="" THEN GOT
   O 2790
HF 2590 TNUM=4:GOSUB 1750:GOSUB 3120
CY 2600 INPUT SPEED4$
HE 2610 GOSUB 3150:INPUT MIN4$
GC 2620 GOSUB 1750:POSITION 2,17:? "Enter
    Your FIFTH Title":FOR WAIT=1 TO 90:NE
   XT WAIT
MJ 2630 TRAP 2620:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE5$:IF TITLE5$="" THEN GOT
   O 2790
HD 2640 TNUM=5:GOSUB 1750:GOSUB 3120
EE 2650 INPUT SPEED5$
IT 2660 GOSUB 3150:INPUT MIN5$
JS 2670 GOSUB 1750:POSITION 2,17:? "Enter
    Your SIXTH Title":FOR WAIT=1 TO 90:NE
   XT WAIT
TW 2680 TRAP 2670:GOSUB 3080:POSITION 1,1
   8:INPUT TITLE6$:IF TITLE6$="" THEN GOT
   O 2790
ID 2690 TNUM=6:GOSUB 1750:GOSUB 3120
EI 2700 INPUT SPEED6$
JG 2710 GOSUB 3150:INPUT MIN6$
CW 2720 IF MODE=2 THEN GOSUB 1750:POSITIO
   N 2,17:? "Enter Your SEVENTH Title":FO
   R WAIT=1 TO 90:NEXT WAIT
VL 2730 IF MODE=2 THEN TRAP 2720:GOSUB 30
   80:POSITION 1,18:INPUT TITLE7$:IF TITL
   E7$="" THEN GOTO 2790
ND 2740 IF MODE=2 THEN TNUM=7:GOSUB 1750:
   GOSUB 3120:INPUT SPEED7$
AQ 2750 IF MODE=2 THEN GOSUB 3150:INPUT M
   IN7$
EE 2790 GOSUB 1750
JU 2795 POSITION 2,16:? "Position LABEL i
   n the printer,then"
ZD 2800 POSITION 10,17:? " TURN ON PRINTE
   R ":FOR J=15 TO 0 STEP -1:SOUND 0,120,
   10,J:NEXT J
EB 2810 FOR WAIT=1 TO 300:NEXT WAIT
KM 2830 IF MODE=1 AND TNUM<=3 THEN LS=12
TP 2835 IF MODE=1 AND TNUM>3 THEN LS=6
MD 2840 IF MODE=2 AND TNUM<=4 THEN LS=12
VF 2845 IF MODE=2 AND TNUM>4 THEN LS=6
NP 2850 LINE=15:GOSUB 1750
RX 2854 IF MODE=1 THEN GOSUB 3050
TW 2856 IF MODE=2 THEN GOSUB 3060
TL 2860 TRAP 1920:POSITION 9,16:? " SET T
   HE LEFT MARGIN ":? "Refer to list for
   suggested setting"
JU 2870 ? "MARGIN=";:INPUT MAR
VF 2880 IF MAR<1 OR MAR>40 THEN 2850
QA 2890 GOTO 3200
FO 3000 POSITION 4,16:? "Refer to list fo
   r the suggested":? "  maximum number o
   f characters"
HK 3010 ? "  (CMAX) allowable for your ti
   tle":RETURN
MM 3015 POKE 82,20:POSITION 20,2:? " Bran
   d":? " ----------------":? " SONY":? :?
    " SCOTCH"
DK 3017 ? :? " FUJI":? :? " MAXELL":POKE
   82,2:RETURN
IA 3020 POKE 82,32:POSITION 31,2:? "CMAX"
   :? :? "31":? :? "20":? :? "31":? :? "3
   0":POKE 82,2:RETURN
IO 3025 POKE 82,20:POSITION 20,2:? " Bran
   d":? " ----------------":? " SCOTCH":? "
    SCOTCH EG":? " MAXELL EX"
HP 3027 ? " MAXELL HGX":? " FUJI":? " MEM
   OREX":? " POLAROID":? " PANASONIC":POK
   E 82,2:RETURN
MF 3030 POKE 82,32:POSITION 31,2:? "CMAX"
   :? :? "35":? "27":? "30":? "27":? "35"
   :? "31"
XS 3040 ? "28":? "30":POKE 82,2:RETURN
RD 3050 POKE 82,31:POSITION 29,2:? "MARGI
   N":? :? " 8 ":? :? "10 ":? :? " 9 ":?
   :? " 8 ":POKE 82,2:RETURN
DY 3060 POKE 82,32:POSITION 30,2:? "MARGI
   N":? :? " 3 ":? " 3 ":? "11 ":? "14 ":
   ? "10 "
EG 3070 ? " 8 ":? "18 ":? "17 ":POKE 82,2
   :RETURN
SW 3080 POSITION 2,17:? "-----------------
   ---------------------"
RB 3090 POSITION 2,16:? "    5    10    15
      20    25    30    35":RETURN
GP 3100 POSITION 2,16:? "Enter Speed (BII
    or BIII):";:RETURN
DX 3110 POSITION 2,16:? "Enter Speed (SP,
   LP or EP):";:RETURN
LY 3120 IF MODE=1 THEN GOSUB 3100
NS 3130 IF MODE=2 THEN GOSUB 3110
AP 3140 RETURN
OW 3150 POSITION 2,17:? "Program Length (
   in minutes)";:RETURN
NG 3199 REM PRINTER CONTROL CODES
FZ 3200 LPRINT CHR$(27);CHR$(64):REM INIT
   IALIZE PRINTER
TW 3210 LPRINT CHR$(27);CHR$(110):REM NLQ
    PICA
GP 3220 LPRINT CHR$(27);CHR$(65);CHR$(LS)
   :REM LS/72 LINE SPACING
JU 3230 LPRINT CHR$(27);CHR$(77);CHR$(MAR
   ):REM LEFT MARGIN
CV 3240 IF MODE=1 AND TNUM>3 THEN LPRINT
   CHR$(27);CHR$(83);CHR$(1):REM SUBSCRIP
   T FONT
FC 3245 IF MODE=2 AND TNUM>4 THEN LPRINT
   CHR$(27);CHR$(83);CHR$(1):REM SUBSCRIP
   T FONT
CB 3250 GOSUB 1750:POSITION 6,17:? "Press
    ANY KEY to print LABEL"
LM 3260 CLOSE #2:OPEN #2,4,0,"K:":GET #2,
   K:CLOSE #2
XR 3270 REM PRINT LABEL
YB 3280 LPRINT TITLE1$;"  ";SPEED1$;" ";M
   IN1$;" "
BQ 3290 LPRINT TITLE2$;"  ";SPEED2$;" ";M
   IN2$;" "
ED 3300 LPRINT TITLE3$;"  ";SPEED3$;" ";M
   IN3$;" "
PK 3310 IF TNUM>3 THEN LPRINT TITLE4$;"
   ";SPEED4$;" ";MIN4$;" "
US 3320 IF TNUM>3 THEN LPRINT TITLE5$;"
   ";SPEED5$;" ";MIN5$;" "
AA 3330 IF TNUM>3 THEN LPRINT TITLE6$;"
   ";SPEED6$;" ";MIN6$;" "
GV 3340 IF TNUM>3 AND MODE=2 THEN LPRINT
   TITLE7$;"  ";SPEED7$;" ";MIN7$;" "
ZM 3350 GOSUB 1710:GOSUB 1750:GOTO 380
```

Red Squares

Article on page 24

LISTING 1

Don't type the TYPO II Codes!

```
XM 10 REM RED SQUARES
LW 20 REM BY MARC ABRAMOWITZ
PB 30 REM (C)1989, ANTIC PUBLISHING INC.
KC 40 REM (DO NOT RENUMBER THIS PROGRAM)
MQ 70 GOTO 120
ZY 80 RETURN
LT 100 BRK=USR(ADR("h)♥┐Mhhl♥P╲)p┐♣▀▄R◆")
   ,1)
HQ 105 POKE 708,54:POKE 711,34:POKE 709,1
   0:RETURN
QJ 120 DIM NAME$(10):GOSUB 8000
AZ 130 POKE 106,PEEK(740)-5:GRAPHICS 17:G
   OSUB 100
AN 135 CHSET=(PEEK(106)+1)*256
TA 140 JUNK=USR(MOV,57344,CHSET,1024)
BQ 160 FOR I=0 TO 7:POKE CHSET+520+I,255:
```

```
   POKE CHSET+8+I,255:POKE CHSET+512+I,0:
   NEXT I
TH 170 DIM X(15):FOR I=0 TO 15:X(I)=80:NE
   XT I:X(7)=3000:X(11)=4000:X(6)=3000
DB 171 X(0)=3000:X(1)=4000
AI 175 X(10)=4000:X(5)=3000:X(9)=4000:X(0
   )=3000:X(1)=4000
ME 180 GOTO 7000
YP 199 REM DRAW BOARD
FK 200 GRAPHICS 17:GOSUB 100:SCORE=0:LINE
   S=0:POKE 756,CHSET/256+2:IF HEIGHT=0 T
   HEN 207
CU 201 FOR Y=17 TO 18-HEIGHT STEP -1:FOR
   X=3 TO 16:R=INT(2*RND(1)+1):IF R=1 THE
   N 206
MK 202 R=INT(4*RND(1)+1):POSITION X,Y:IF
   R=1 THEN ? #6;"!"
PR 203 IF R=2 THEN ? #6;"■"
EQ 204 IF R=3 THEN ? #6;"■"
OJ 205 IF R=4 THEN ? #6;"■"
UY 206 NEXT X:NEXT Y
GP 207 FOR Y=0 TO 18:POSITION 0,Y:? #6;"
   ■":POSITION 17,Y:? #6;"■  ":NEXT Y
TL 208 FOR X=2 TO 17:POSITION X,18:? #6;"
   ■":NEXT X
EY 209 POSITION 2,18:? #6;"■":POSITION 17
   ,18:? #6;"■"
AX 210 X=10:Y=0:SHAPE=INT(RND(0)*7)+1:POK
   E 77,0
XP 220 Y=Y+1
WD 230 TR=STRIG(0):KEY=PEEK(764)
NK 240 GOSUB X(PEEK(632)):IF KEY<2 THEN G
   OSUB X(KEY)
MW 255 IF (TR=0 OR KEY=5) AND Y>3 THEN GO
   SUB 2000
DG 260 POSITION X,Y:POKE 764,255
NL 270 IF SHAPE=1 THEN ? #6;"!■■■":GOTO 3
   70
EV 280 IF SHAPE=2 THEN ? #6;"!■":POSITION
    X,Y-1:? #6;"■■":GOTO 370
GN 290 IF SHAPE=3 THEN ? #6;"!■■":POSITIO
   N X,Y-1:? #6;"■":GOTO 370
XE 300 IF SHAPE=4 THEN ? #6;"!■■":POSITIO
   N X+2,Y-1:? #6;"■":GOTO 370
UT 310 IF SHAPE=5 THEN ? #6;"!■":POSITION
    X-1,Y-1:? #6;"■■":GOTO 370
SF 320 IF SHAPE=6 THEN ? #6;"!■":POSITION
    X+1,Y-1:? #6;"■■":GOTO 370
PJ 322 IF SHAPE=7 THEN ? #6;"!":POSITION
   X-1,Y-1:? #6;"■■■":GOTO 370
TU 323 IF SHAPE=8 AND Y>2 THEN GOSUB 1300
   :GOTO 370
JL 324 IF SHAPE=9 AND Y>1 THEN ? #6;"!■":
   POSITION X+1,Y-1:? #6;"■":POSITION X+1
   ,Y-2:? #6;"■":GOTO 370
BD 325 IF SHAPE=10 AND Y>1 THEN ? #6;"!":
   POSITION X,Y-1:? #6;"■":POSITION X-1,Y
   -2:? #6;"■■":GOTO 370
AU 326 IF SHAPE=11 AND Y>1 THEN ? #6;"!":
   POSITION X,Y-1:? #6;"■■":POSITION X+1,
   Y-2:? #6;"■":GOTO 370
WN 327 IF SHAPE=12 AND Y>1 THEN ? #6;"!":
   POSITION X-1,Y-1:? #6;"■■":POSITION X-
   1,Y-2:? #6;"■":GOTO 370
DT 328 IF SHAPE=13 AND Y>1 THEN ? #6;"!":
   POSITION X,Y-1:? #6;"■■":POSITION X,Y-
   2:? #6;"■":GOTO 370
GC 329 IF SHAPE=14 THEN ? #6;"!":POSITION
    X-2,Y-1:? #6;"■■■":GOTO 370
CX 330 IF SHAPE=15 THEN ? #6;"!":POSITION
    X,Y-1:? #6;"■■■":GOTO 370
MW 331 IF SHAPE=16 THEN ? #6;"!■■":POSITI
   ON X+1,Y-1:? #6;"■":GOTO 370
EI 332 IF SHAPE=17 AND Y>1 THEN ? #6;"!":
   POSITION X,Y-1:? #6;"■":POSITION X,Y-2
   :? #6;"■■":GOTO 370
VR 333 IF SHAPE=18 AND Y>1 THEN ? #6;"!■"
   :POSITION X,Y-1:? #6;"■":POSITION X,Y-
   2:? #6;"■":GOTO 370
GD 334 IF SHAPE=19 AND Y>1 THEN ? #6;"!":
   POSITION X-1,Y-1:? #6;"■■":POSITION X,
   Y-2:? #6;"■"
WE 370 FOR D=1 TO SPEED:NEXT D
JG 500 Z1=32:Z2=32:Z3=32:Z4=32:LOCATE X,Y
   +1,Z1
RY 510 IF SHAPE=1 THEN LOCATE X+1,Y+1,Z2:
   LOCATE X+2,Y+1,Z3:LOCATE X+3,Y+1,Z4:GO
   TO 700
ER 520 IF SHAPE=2 THEN LOCATE X+1,Y+1,Z2:
   GOTO 700
LN 530 IF SHAPE=3 THEN LOCATE X+1,Y+1,Z2:
   LOCATE X+2,Y+1,Z3:GOTO 700
MD 540 IF SHAPE=4 THEN LOCATE X+1,Y+1,Z2:
   LOCATE X+2,Y+1,Z3:GOTO 700
JI 550 IF SHAPE=5 THEN LOCATE X+1,Y+1,Z2:
   LOCATE X-1,Y,Z3:GOTO 700
IE 560 IF SHAPE=6 THEN LOCATE X+1,Y+1,Z2:
   LOCATE X+2,Y,Z3:GOTO 700
OR 570 IF SHAPE=7 THEN LOCATE X-1,Y,Z2:LO
   CATE X+1,Y,Z3:GOTO 700
IZ 590 IF SHAPE=9 THEN LOCATE X+1,Y+1,Z2:
   GOTO 700
IW 600 IF SHAPE=10 THEN LOCATE X-1,Y-1,Z2
   :GOTO 700
HQ 610 IF SHAPE=11 THEN LOCATE X+1,Y,Z2:G
   OTO 700
KP 620 IF SHAPE=12 THEN LOCATE X-1,Y,Z2:G
   OTO 700
IZ 621 IF SHAPE=13 THEN LOCATE X+1,Y,Z2:G
   OTO 700
CZ 622 IF SHAPE=14 THEN LOCATE X-1,Y,Z2:L
   OCATE X-2,Y,Z3:GOTO 700
XV 623 IF SHAPE=15 THEN LOCATE X+1,Y,Z2:L
   OCATE X+2,Y,Z3:GOTO 700
FJ 624 IF SHAPE=16 THEN LOCATE X+1,Y+1,Z2
   :LOCATE X+2,Y+1,Z3:GOTO 700
LI 625 IF SHAPE=17 THEN LOCATE X+1,Y-1,Z2
   :GOTO 700
JK 626 IF SHAPE=18 THEN LOCATE X+1,Y+1,Z2
   :GOTO 700
HW 627 IF SHAPE=19 THEN LOCATE X-1,Y,Z2
DU 700 IF Z1<>32 OR Z2<>32 OR Z3<>32 OR Z
   4<>32 THEN POP :SCORE=SCORE+1:GOSUB 50
   00:GOTO 207
PG 1000 POSITION X,Y:IF SHAPE=1 THEN ? #6
   ;"    "
UY 1010 IF SHAPE=2 THEN ? #6;"  ":POSITIO
   N X,Y-1:? #6;"  "
VH 1020 IF SHAPE=3 THEN ? #6;"   ":POSITI
   ON X,Y-1:? #6;" "
SA 1030 IF SHAPE=4 THEN ? #6;"   ":POSITI
   ON X+2,Y-1:? #6;" "
TN 1040 IF SHAPE=5 THEN ? #6;"  ":POSITIO
   N X-1,Y-1:? #6;"  "
QZ 1050 IF SHAPE=6 THEN ? #6;"  ":POSITIO
   N X+1,Y-1:? #6;"  "
UC 1060 IF SHAPE=7 THEN ? #6;" ":POSITION
    X-1,Y-1:? #6;"   "
UM 1061 IF SHAPE=8 AND Y>2 THEN ? #6;" ":
   POSITION X,Y-1:? #6;" ":POSITION X,Y-2
   :? #6;" ":POSITION X,Y-3:? #6;" "
DO 1062 IF SHAPE=9 AND Y>1 THEN ? #6;"  "
   :POSITION X+1,Y-1:? #6;" ":POSITION X+
   1,Y-2:? #6;" "
GA 1063 IF SHAPE=10 AND Y>1 THEN ? #6;" "
   :POSITION X,Y-1:? #6;" ":POSITION X-1,
   Y-2:? #6;"  "
BW 1064 IF SHAPE=11 AND Y>1 THEN ? #6;" "
   :POSITION X,Y-1:? #6;"  ":POSITION X+1
   ,Y-2:? #6;" "
VN 1065 IF SHAPE=12 AND Y>1 THEN ? #6;" "
   :POSITION X-1,Y-1:? #6;"  ":POSITION X
   -1,Y-2:? #6;" "
ZT 1066 IF SHAPE=13 AND Y>1 THEN ? #6;" "
   :POSITION X,Y-1:? #6;"  ":POSITION X,Y
   -2:? #6;" "
JY 1067 IF SHAPE=14 THEN ? #6;" ":POSITIO
   N X-2,Y-1:? #6;"   "
KC 1068 IF SHAPE=15 THEN ? #6;" ":POSITIO
   N X,Y-1:? #6;"   "
IB 1069 IF SHAPE=16 THEN ? #6;"   ":POSIT
   ION X+1,Y-1:? #6;" "
BS 1070 IF SHAPE=17 AND Y>1 THEN ? #6;" "
   :POSITION X,Y-1:? #6;" ":POSITION X,Y-
   2:? #6;"  "
BV 1071 IF SHAPE=18 AND Y>1 THEN ? #6;"
   ":POSITION X,Y-1:? #6;" ":POSITION X,Y
   -2:? #6;" "
IV 1072 IF SHAPE=19 AND Y>1 THEN ? #6;" "
   :POSITION X-1,Y-1:? #6;"  ":POSITION X
   ,Y-2:? #6;" "
OF 1200 GOTO 220
EO 1300 ? #6;"!":POSITION X,Y-1:? #6;"■":
   POSITION X,Y-2:? #6;"■":POSITION X,Y-3
   :? #6;"■":RETURN
MA 2000 IF SHAPE=1 THEN SHAPE=8:RETURN
KF 2005 IF SHAPE=2 AND X<16 THEN RETURN
RT 2010 IF SHAPE=3 AND X<16 THEN SHAPE=9:
   RETURN
WM 2011 IF SHAPE=4 AND X>3 THEN SHAPE=10:
   RETURN
WD 2012 IF SHAPE=5 AND X<16 THEN SHAPE=11
```

```
   :RETURN
AU 2013 IF SHAPE=6 AND X>3 THEN SHAPE=12:RETURN
AN 2014 IF SHAPE=7 AND X<16 THEN SHAPE=13:RETURN
HN 2020 IF SHAPE=8 AND X<14 THEN SHAPE=1:RETURN
PH 2030 IF SHAPE=11 AND X>3 AND X<16 THEN SHAPE=5:RETURN
NC 2040 IF SHAPE=12 AND X<16 THEN SHAPE=6:RETURN
GG 2050 IF SHAPE=9 AND X>4 THEN SHAPE=14:RETURN
AA 2060 IF SHAPE=10 AND X<15 THEN SHAPE=15:RETURN
DM 2070 IF SHAPE=13 AND X<15 THEN SHAPE=16:RETURN
GR 2080 IF SHAPE=14 AND X<16 THEN SHAPE=17:RETURN
IX 2090 IF SHAPE=15 AND X<16 THEN SHAPE=18:RETURN
KI 2100 IF SHAPE=16 AND X>3 THEN SHAPE=19:RETURN
KO 2110 IF SHAPE=17 AND X<15 THEN SHAPE=3:RETURN
MT 2120 IF SHAPE=18 AND X<15 THEN SHAPE=4:RETURN
VB 2130 IF SHAPE=19 AND X>3 THEN SHAPE=7:RETURN
BQ 2980 RETURN
UK 3000 Z1=32:Z2=32:Z3=32
SR 3010 IF SHAPE=1 THEN LOCATE X+4,Y,Z1:GOTO 3890
YC 3020 IF SHAPE=2 THEN LOCATE X+2,Y,Z1:LOCATE X+2,Y-1,Z2:GOTO 3890
SW 3030 IF SHAPE=3 THEN LOCATE X+3,Y,Z1:GOTO 3890
XS 3040 IF SHAPE=4 THEN LOCATE X+3,Y-1,Z1:LOCATE X+3,Y,Z2:GOTO 3890
YJ 3050 IF SHAPE=5 THEN LOCATE X+2,Y,Z1:LOCATE X+1,Y-1,Z2:GOTO 3890
XF 3060 IF SHAPE=6 THEN LOCATE X+3,Y-1,Z1:LOCATE X+2,Y,Z2:GOTO 3890
BA 3065 IF SHAPE=7 THEN LOCATE X+1,Y,Z1:LOCATE X+2,Y-1,Z2:GOTO 3890
RV 3066 IF SHAPE=8 THEN LOCATE X+1,Y,Z1:LOCATE X+1,Y-1,Z2:LOCATE X+1,Y-2,Z3:LOCATE X+1,Y-3,Z4:GOTO 3890
MZ 3067 IF SHAPE=9 THEN LOCATE X+2,Y,Z1:LOCATE X+2,Y-1,Z2:LOCATE X+2,Y-2,Z3:GOTO 3890
LW 3068 IF SHAPE=10 THEN LOCATE X+1,Y,Z1:LOCATE X+1,Y-1,Z2:LOCATE X+1,Y-2,Z3:GOTO 3890
RA 3069 IF SHAPE=11 THEN LOCATE X+1,Y,Z1:LOCATE X+2,Y-1,Z2:LOCATE X+2,Y-2,Z3:GOTO 3890
NA 3070 IF SHAPE=12 THEN LOCATE X+1,Y,Z1:LOCATE X+1,Y-1,Z2:LOCATE X,Y-2,Z3:GOTO 3890
OP 3071 IF SHAPE=13 THEN LOCATE X+1,Y,Z1:LOCATE X+2,Y-1,Z2:LOCATE X+1,Y-2,Z3:GOTO 3890
PN 3072 IF SHAPE=14 THEN LOCATE X+1,Y,Z1:LOCATE X+1,Y-1,Z2:GOTO 3890
TZ 3073 IF SHAPE=15 THEN LOCATE X+1,Y,Z1:LOCATE X+3,Y-1,Z2:GOTO 3890
VJ 3074 IF SHAPE=16 THEN LOCATE X+3,Y,Z1:LOCATE X+2,Y-1,Z2:GOTO 3890
SJ 3075 IF SHAPE=17 THEN LOCATE X+1,Y,Z1:LOCATE X+1,Y-1,Z2:LOCATE X+2,Y-2,Z3:GOTO 3890
RV 3076 IF SHAPE=18 THEN LOCATE X+2,Y,Z1:LOCATE X+1,Y-1,Z2:LOCATE X+1,Y-2,Z3:GOTO 3890
PE 3077 IF SHAPE=19 THEN LOCATE X+1,Y,Z1:LOCATE X+1,Y-1,Z2:LOCATE X+1,Y-2,Z3
UK 3890 IF Z1=32 AND Z2=32 AND Z3=32 THEN X=X+1
AT 3900 RETURN
VE 4000 Z1=32:Z2=32:Z3=32:LOCATE X-1,Y,Z1
RJ 4010 IF SHAPE=1 THEN LOCATE X-1,Y,Z1:GOTO 4880
XS 4020 IF SHAPE=2 THEN LOCATE X-1,Y-1,Z2:GOTO 4880
YK 4030 IF SHAPE=3 THEN LOCATE X-1,Y-1,Z2:GOTO 4880
WU 4040 IF SHAPE=4 THEN LOCATE X+1,Y-1,Z2:GOTO 4880
AZ 4050 IF SHAPE=5 THEN LOCATE X-2,Y-1,Z2:GOTO 4880
TF 4060 IF SHAPE=6 THEN LOCATE X,Y-1,Z2:GOTO 4880
DA 4065 IF SHAPE=7 THEN LOCATE X-2,Y-1,Z2:GOTO 4880
XV 4066 IF SHAPE=8 THEN LOCATE X-1,Y-1,Z2:LOCATE X-1,Y-2,Z3:LOCATE X-1,Y-3,Z4:GOTO 4880
VQ 4067 IF SHAPE=9 THEN LOCATE X,Y-1,Z2:LOCATE X,Y-2,Z3:GOTO 4880
RS 4068 IF SHAPE=10 THEN LOCATE X-1,Y-1,Z2:LOCATE X-2,Y-2,Z3:GOTO 4880
SL 4069 IF SHAPE=11 THEN LOCATE X-1,Y-1,Z2:LOCATE X,Y-2,Z3:GOTO 4880
TB 4070 IF SHAPE=12 THEN LOCATE X-2,Y-1,Z2:LOCATE X-2,Y-2,Z3:GOTO 4880
QR 4071 IF SHAPE=13 THEN LOCATE X-1,Y-1,Z2:LOCATE X-1,Y-2,Z3:GOTO 4880
HF 4072 IF SHAPE=14 THEN LOCATE X-3,Y-1,Z2:GOTO 4880
FN 4073 IF SHAPE=15 THEN LOCATE X-1,Y-1,Z2:GOTO 4880
VK 4074 IF SHAPE=16 THEN LOCATE X,Y-1,Z2:GOTO 4880
TT 4075 IF SHAPE=17 THEN LOCATE X-1,Y-1,Z2:LOCATE X-1,Y-2,Z3:GOTO 4880
UN 4076 IF SHAPE=18 THEN LOCATE X-1,Y-1,Z2:LOCATE X-1,Y-2,Z3:GOTO 4880
IW 4077 IF SHAPE=19 THEN LOCATE X-2,Y-1,Z2:LOCATE X-1,Y-2,Z3
XQ 4880 IF Z1=32 AND Z2=32 AND Z3=32 THEN X=X-1
BT 4890 RETURN
FF 5000 IF Y<=1 THEN POP :GOTO 6000
ID 5010 REM
DL 5020 FOR Y=17 TO 0 STEP -1
ZP 5030 X=3
GJ 5040 LOCATE X,Y,Z1
ZE 5050 IF Z1=32 THEN 5080
CU 5060 IF X<16 THEN X=X+1:GOTO 5040
RM 5070 FOR S=2 TO 17:POKE 712,(S*8)+4:SOUND 0,S*10,0,(17-S):NEXT S:POKE 712,0
DB 5071 SOUND 0,0,0,0:POSITION 3,Y:? #6;"              ";
ZL 5072 FOR Y2=Y TO 1 STEP -1:ZF=0:FOR X=3 TO 16:LOCATE X,Y2-1,Z1:IF Z1=32 THEN ZF=ZF+1
CR 5073 IF ZF=14 THEN POP :POP :GOTO 5079
UZ 5074 POSITION X,Y2:? #6;CHR$(Z1):POSITION X,Y2-1:? #6;" "
ZD 5075 NEXT X:NEXT Y2
SP 5079 LINES=LINES+1:SCORE=SCORE+10*(18-Y):SPEED=SPEED-5:GOTO 5030
ML 5080 NEXT Y
BE 5090 RETURN
BG 5959 IF Z1=32 THEN 5080
ID 6000 GRAPHICS 18:GOSUB 100:LINE$="game over":X=4:Y=4:GOSUB 7170
QX 6010 FOR D=1 TO 500:NEXT D
IE 6020 LINE$="LINES:   ":LINE$(LEN(LINE$)+1)=STR$(LINES):Y=5:LINE$(LEN(LINE$)+1)="   ":GOSUB 7170
UV 6030 LINE$="SCORE:   ":LINE$(LEN(LINE$)+1)=STR$(SCORE):Y=4:GOSUB 7170
IX 6040 LINE$="press start":LL=5:RL=7:X=3:Y=7
AX 6050 GOSUB 7200
UO 6090 POKE 709,PEEK(20):IF PEEK(53279)<>6 AND STRIG(0)=1 THEN 6090
SI 6100 GOTO 7150
OT 7000 GRAPHICS 18:GOSUB 100:DIM LINE$(20)
PV 7010 LINE$="RED SQUARES":X=3:Y=4:GOSUB 7170
EI 7040 LINE$="by marc abramowitz":X=0:Y=6:LL=9:RL=10:GOSUB 7200
QN 7050 LINE$="press start"
CY 7120 FOR L=1 TO 11:FOR F=9 TO 8 STEP -1:POSITION 3+L,F+1:? #6;" ":SOUND 0,F*25,10,8:POSITION 3+L,F
RF 7130 ? #6;LINE$(L,L):NEXT F:NEXT L
HQ 7140 SOUND 0,0,0,0
XH 7150 POKE 711,PEEK(20):IF PEEK(53279)<>6 AND STRIG(0) THEN 7150
ZP 7155 GOSUB 9000:GOTO 200
LQ 7170 FOR L=1 TO LEN(LINE$):POSITION X+L,Y:FOR F=15 TO 0 STEP -5
HK 7175 SOUND 0,F*(255/Y),10,F:NEXT F:? #6;LINE$(L,L):NEXT L
YG 7180 SOUND 0,0,0,0:RETURN
ZM 7200 FOR I=1 TO X+LL:POSITION I-1,Y:?
```

```
   #6;" ":SOUND 0,I*10,10,8:POSITION I,Y:
   ? #6;LINE$(LL,LL):NEXT I
UX 7210 FOR I=18 TO X+RL STEP -1:POSITION
    I+1,Y:? #6;" ":SOUND 0,I*10,10,8:POSI
   TION I,Y:? #6;LINE$(RL,RL):NEXT I
JF 7220 SOUND 0,0,0,0:LL=LL-1:RL=RL+1:IF
   LL<=0 THEN RETURN
QT 7230 GOTO 7200
NB 8000 REM * ML MOVE ROUTINE *
IV 8010 MOV=ADR("hh9h9Ph9Uh9Th9Ohh9n■♥LPP
   TfTPLfUfPPLfGfnPJfO*f◆")
AM 8020 RETURN
OV 9000 GRAPHICS 2:GOSUB 100:SPEED=1:HEIG
   HT=0:? :POKE 657,3:? "USE CONSOLE KEYS
    OR JOYSTICK TO"
JA 9005 POKE 657,13:? "SELECT OPTIONS":PO
   KE 710,0:POKE 755,0
ZD 9010 POSITION 1,5:? #6;"select - SPEED
   "
IM 9020 POSITION 1,6:? #6;"option - HEIGH
   T"
ZN 9025 POSITION 1,7:? #6;"start - BEGIN
   GAME"
RD 9030 POSITION 1,2:? #6;"SPEED (1-50):
    ";SPEED;" "
TW 9040 POSITION 1,3:? #6;"HEIGHT (0-10):
    ";HEIGHT;" "
HM 9050 ST=PEEK(632):CON=PEEK(53279)
XJ 9051 IF PEEK(53279)=6 OR STRIG(0)=0 TH
   EN 9150
ZO 9052 IF ST=15 AND CON=7 AND STRIG(0)=1
    THEN 9050
QZ 9054 FOR V=15 TO 0 STEP -1.5:SOUND 0,5
   0,10,V:NEXT V
LT 9060 IF ST=14 OR CON=5 THEN SPEED=SPEE
   D+1
RW 9070 IF ST=13 THEN SPEED=SPEED-1
SY 9080 IF ST=7 OR CON=3 THEN HEIGHT=HEIG
   HT+1
WU 9090 IF ST=11 THEN HEIGHT=HEIGHT-1
BB 9100 IF SPEED>50 THEN SPEED=1
TO 9110 IF SPEED<1 THEN SPEED=50
JV 9120 IF HEIGHT>10 THEN HEIGHT=0
ZX 9130 IF HEIGHT<0 THEN HEIGHT=10
RH 9140 GOTO 9010
SQ 9150 SPEED=50-SPEED:RETURN
```

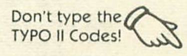
LISTING 2

```
AY 10 REM
AZ 20 REM
GD 30 REM (c) 1985,1988 ANTIC PUBLISHING
EV 40 REM (LINES 10-250 MAY BE USED WITH
   OTHER BASIC LOADERS IN THIS ISSUE.
IJ 50 REM CHANGE LINE 70 AS NECESSARY.)
PR 60 DIM FN$(20),TEMP$(20),AR$(93):DPL=P
   EEK(10592):POKE 10592,255
WO 70 FN$="D:LINES.LST":REM THIS IS THE N
   AME OF THE DISK FILE TO BE CREATED
RD 80 ? "Disk or Cassette?";:POKE 764,25
   5
PY 90 IF  NOT (PEEK(764)=18 OR PEEK(764)=
   58) THEN 90
TH 100 IF PEEK(764)=18 THEN FN$="C:"
VB 110 POKE 764,255:GRAPHICS 0:? "      AN
   TIC'S GENERIC BASIC LOADER"
MY 120 ? ,"BY CHARLES JACKSON"
KB 130 POKE 10592,DPL:TRAP 200
PU 140 ? :? :? "Creating ";FN$:? "...plea
   se stand by."
LW 150 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BQ 160 AR$="":READ AR$
YC 170 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DM 180 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")      "
BK 190 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 160
MM 200 IF PEEK(195)=5 THEN ? :? :? "TOO
   MANY DATA LINES!":? "CANNOT CREATE FIL
   E!":END
CM 210 IF C<LN+1 THEN ? :? "TOO FEW DATA
    LINES!":? "CANNOT CREATE FILE!":END
UQ 220 IF FN$="C:" THEN ? :? " Prepare ca
   ssette, press [RETURN]"
AR 230 OPEN #1,8,0,FN$
PV 240 POKE 766,1:? #1;A$;:POKE 766,0
AL 250 CLOSE #1:GRAPHICS 0:? " COMPLETED 
   "
FX 1000 DATA 105
EF 1010 DATA 0490480480320660820750610850
   8308204000650680820400341041690001330 77
   104104201000240007169112133
TZ 1020 DATA 0161410142100960340410440490
   4115505604804904803207707908606106506 8
   082040003410410413324110413 3
UT 1030 DATA 2401041332131041332121041332
   3910413323816000017724014521223021220 8
   0022302132302402080022302 41
RE 1040 DATA 1982382082341982390162300960
   34041155
```

Flash!

Article on page 25

LISTING 1




```
AY 10 REM
AZ 20 REM
GD 30 REM (c) 1985,1988 ANTIC PUBLISHING
EV 40 REM (LINES 10-250 MAY BE USED WITH
   OTHER BASIC LOADERS IN THIS ISSUE.
IJ 50 REM CHANGE LINE 70 AS NECESSARY.)
PR 60 DIM FN$(20),TEMP$(20),AR$(93):DPL=P
   EEK(10592):POKE 10592,255
EA 70 FN$="D:FLASH.EXE":REM THIS IS THE N
   AME OF THE DISK FILE TO BE CREATED
RD 80 ? "Disk or Cassette?";:POKE 764,25
   5
PY 90 IF  NOT (PEEK(764)=18 OR PEEK(764)=
   58) THEN 90
TH 100 IF PEEK(764)=18 THEN FN$="C:"
VB 110 POKE 764,255:GRAPHICS 0:? "      AN
   TIC'S GENERIC BASIC LOADER"
MY 120 ? ,"BY CHARLES JACKSON"
KB 130 POKE 10592,DPL:TRAP 200
PU 140 ? :? :? "Creating ";FN$:? "...plea
   se stand by."
LW 150 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BQ 160 AR$="":READ AR$
YC 170 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DM 180 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")      "
BK 190 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 160
MM 200 IF PEEK(195)=5 THEN ? :? :? "TOO
   MANY DATA LINES!":? "CANNOT CREATE FIL
   E!":END
CM 210 IF C<LN+1 THEN ? :? "TOO FEW DATA
    LINES!":? "CANNOT CREATE FILE!":END
UQ 220 IF FN$="C:" THEN ? :? " Prepare ca
   ssette, press [RETURN]"
AR 230 OPEN #1,8,0,FN$
PV 240 POKE 766,1:? #1;A$;:POKE 766,0
AL 250 CLOSE #1:GRAPHICS 0:? " COMPLETED 
   "
```

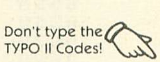
```
LD 1000 DATA 945
GD 1010 DATA 2552550000080251080169064133213169000133212169224133205169000133204133240162007160000177204153
CN 1020 DATA 0001122380250802080003238026080157208000623020420800022302052022242552082290320960801652120241 05
VP 1030 DATA 0081332121652131050001332132302401652402011282082010760210810011280421550000000000000000000000
NG 1040 DATA 0002551291291291291291292551690001332031600001620000241652032400010561852080006042153208006169
TQ 1050 DATA 0001440021690011332031260320062322240082082272001920082082201600001850320060732551452122001 92
PM 1060 DATA 0082082440961690721332131690001332121690641332051690001332041691001332411690001332401600001 77
RR 1070 DATA 2122010321760051050640241440262010961760060562330320241440162011281440122011601760021442312 01
NA 1080 DATA 2241760021442341600001322431332421600030062420382431362082491652430241050641332431600001772 42
GM 1090 DATA 0732551452402001920082082451652520802470812400241050081332401652411050001332412302122080022 30
YY 1100 DATA 2131652132010732081570960760530811250290290320320320320320320320320320320320320700321080320 97
NP 1110 DATA 0321150321040320330320331551620001690091570660031690241570680031690811570690031690291570720 03
UZ 1120 DATA 1690001570730030320862280761210810290320320651100321011201121011141051091011101160321051100 32
QD 1130 DATA 1161041010321151170981081051091051100971081551620001690091570660031690861570680031690811570 69
UA 1140 DATA 0031690351570720031690001570730030320862280761860810290320320320320320400990410320490570560 57
PP 1150 DATA 0320651101161050990320801170981081051151041051101031551620001690091570660031691541570680031 69
```

```
LG 1160 DATA 081157069003169032157072003169000157073003032086228076246081029032032032032032032032032032
PW 1170 DATA 066121058032069114110105101032078101103117115155162000248081243082169009157066003169219157068
KB 1180 DATA 003169081157069003169027157072003169000157073003032086228076040082029029069110116101114032097
NZ 1190 DATA 032112104114097115101155162000169009157066003169023157068003169082157069003169017157072003169
UO 1200 DATA 000157073003032086228169001141240002162000169011157066003169078157068003169080157069003169001
SU 1210 DATA 157072003169000157073003032086228162000169005157066003169000157068003169072157069003169255157
DH 1220 DATA 072003169000157073003032086228173072003133245076157082125029029029029029029029029029029029
HS 1230 DATA 155162000169009157066003169143157068003169082157069003169014157072003169000157073003032086228
AN 1240 DATA 198245162000189080080157000112232224016208245169112141244002169000141200002141198002169015141
OD 1250 DATA 197002032153080169000133246169100133205169000133204141001006165088024105220133244082152083240
SW 1260 DATA 165089105000133241160000177204162000142000006162000010133203144001232189076080145240165240024
DI 1270 DATA 105040133240165241105000133241165203174000006232224008208217173252002201255240011169000141240
QV 1280 DATA 002169224141244002096173070083240010169000133020165020201001208250230204208002230205238001006
HZ 1290 DATA 032110083173001006201008208146169000141001006230246165246197245208133141031208076223082173031
MU 1300 DATA 208201005208013174070083224060240006238070083076145083201003208017174070083224000240010206070
EP 1310 DATA 083173031208201006144249096224000222500200000 80
```

Machine Language Stringer

Article on page 8

LISTING 1




```
FR 10 REM MACHINE LANGUAGE STRINGER
TU 20 REM by Andy Barton
AD 30 REM (c)1989, ANTIC PUBLISHING INC.
BC 50 REM
QT 60 DIM A$(120),B$(15),C$(120),ML$(1000
   0),S$(5),N$(2):Q=1:ED=0:N=1
ZW 70 C$(1)=" ":C$(120)=" ":C$(2)=C$
AO 80 ? "↖":? :? :? "CONVERT OBJ FILE TO
   STRING DATA":? :?
UJ 90 ? "INPUT OBJ FILE  (Dn:xxx)"
GX 100 INPUT B$:TRAP 90:IF B$(1,1)<>"D" T
   HEN 90
SA 110 CLOSE #1:OPEN #1,4,0,B$
EH 120 ? "STARTING LINE NO."
WV 130 TRAP 120:INPUT SLN:DSLN=SLN:SLN=SL
   N+1:TRAP 40000
FS 135 IF SLN<482 THEN ? :? "STARTING LIN
   E NO. MUST BE GREATER THEN 480":? :G
   OTO 120
FQ 140 ? "NAME ML STRING  (2 CHARACTERS O
   NLY)":INPUT N$:IF LEN(N$)=0 THEN 140
CJ 150 S$=N$:S$(LEN(N$)+1)=STR$(N)
SK 160 ? "↖":TRAP 410
BB 170 GET #1,X:GET #1,X:REM DISCARD FILE
    IDENTIFICATION CODE (255,255)
RG 180 TRAP 420:REM NOMAL EOF
QB 190 GET #1,A:GET #1,B:ST=B*256+A
XR 200 IF ED=0 THEN BST=ST:GOTO 220
IF 210 IF ST<>EDST THEN GOSUB 450:BST=ST:
   N=N+1:S$(LEN(N$)+1)=STR$(N):Q=1:SLN=SL
   N+2:DSLN=SLN:SLN=SLN+1
OK 220 GET #1,A:GET #1,B:ED=B*256+A
ZU 230 POKE 766,1:POSITION 2,4:? C$:POSIT
   ION 2,4:? SLN;" ";S$;"$(";Q;",";:POSIT
   ION 22,4:? ")=";CHR$(34);
AV 240 TRAP 390:REM EARLY EOF
GF 250 FOR Z=1 TO 90:GET #1,X
ET 260 IF X=155 OR X=34 THEN 350
UJ 270 ? CHR$(X);
KE 280 ST=ST+1:IF ST>ED THEN 330
MY 290 NEXT Z
YV 300 POSITION 18,4:? Q+Z-2:Q=Q+90:SLN=S
   LN+1:GOSUB 310:GOTO 230
IS 310 POKE 766,0:POSITION 2,10:? "CONT":
   POSITION 1,2:? " ";:POKE 842,13:STOP
SI 320 POKE 842,12:RETURN
SV 330 REM ST -> ED REACHED
OA 340 POSITION 18,4:? Q+Z-1:GOSUB 310:ED
   ST=ST:Q=Q+Z:SLN=SLN+1:GOTO 180
CY 350 IF Z=1 THEN 360
XG 355 POSITION 18,4:? Q+Z-2:GOSUB 310:SL
   N=SLN+1
UG 360 POSITION 2,4:? C$:POSITION 2,4:? S
   LN;" ";S$;"$(";Q+Z-1;",";Q+Z-1;")=";"C
   HR$(";X;")"
ZM 370 GOSUB 310:SLN=SLN+1:Q=Q+Z:ST=ST+1:
   IF ST>ED THEN EDST=ST:GOTO 180
```

```
NP 380 GOTO 230
NM 390 REM ERROR -- EARLY EOF
RX 400 GOSUB 310
BA 410 ? " ERROR -- EARLY EOF !!!":STOP
MJ 420 GOSUB 310:GOSUB 450:? :? "DONE":PO
   KE 766,0:REM NORMAL EOF
OL 430 ? "■LIST■these lines into a disk f
   ile      then ■ENTER■ them into your bas
   ic       program"
OB 440 END
HY 450 NUM=ED-BST+1:POSITION 2,4:? C$:POS
   ITION 2,4:? DSLN;"DIM ";S$;"$(";NUM;")
   ":GOSUB 310:POSITION 2,4:? C$
IJ 460 POSITION 2,4:? SLN;" X=USR(ADR(";C
   HR$(34);"hh9h Ph Uh Th 0h np fo  1prT
   fTP fUfPP fqFnPIFoPh ";CHR$(34);
BY 470 ? "),ADR(";S$;"$),";BST;",";NUM;")
   "
JF 480 GOSUB 310:RETURN
```

LISTING 2

```
AY 10 REM
AZ 20 REM
GD 30 REM (c) 1985,1988 ANTIC PUBLISHING
EV 40 REM (LINES 10-250 MAY BE USED WITH
   OTHER BASIC LOADERS IN THIS ISSUE.
IJ 50 REM CHANGE LINE 70 AS NECESSARY.)
PR 60 DIM FN$(20),TEMP$(20),AR$(93):DPL=P
   EEK(10592):POKE 10592,255
WO 70 FN$="D:LINES.LST":REM THIS IS THE N
   AME OF THE DISK FILE TO BE CREATED
RD 80 ? "■Disk or ■assette?";:POKE 764,25
   5
PY 90 IF  NOT (PEEK(764)=18 OR PEEK(764)=
   58) THEN 90
TH 100 IF PEEK(764)=18 THEN FN$="C:"
VB 110 POKE 764,255:GRAPHICS 0:?  "      AN
   TIC'S GENERIC BASIC LOADER"
MY 120 ? ,"BY CHARLES JACKSON"
KB 130 POKE 10592,DPL:TRAP 200
PU 140 ? :? :? "Creating ";FN$:? "...plea
   se stand by."
LW 150 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BQ 160 AR$="":READ AR$
YC 170 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DM 180 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")      "
BK 190 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 160
MM 200 IF PEEK(195)=5 THEN ? :? :? "■TOO
   MANY DATA LINES!":? "CANNOT CREATE FIL
   E!":END
CM 210 IF C<LN+1 THEN ? :? "■TOO FEW DATA
    LINES!":? "CANNOT CREATE FILE!":END
UQ 220 IF FN$="C:" THEN ? :? " Prepare ca
   ssette, press [RETURN]"
AR 230 OPEN #1,8,0,FN$
PV 240 POKE 766,1:? #1;A$;:POKE 766,0
AL 250 CLOSE #1:GRAPHICS 0:? "■COMPLETED■
   "
HX 1000 DATA 109
ED 1010 DATA 052054048032080007908307308 40
```

```
HX 1000 DATA 109
ED 1010 DATA 05205404803208000790830730840
   73079078032050044052058063032083076078
   059034032088061085083082040
UH 1020 DATA 0650680820400340590670720820
   36040051052041059034104104133241104133
   240104133213104133212104133
BJ 1030 DATA 2391041332382400022302391600
   00177240145212230212208002230213230240
   208002230241198238208236198
VM 1040 DATA 2392082320960340590670720820
   36040051052041059155
```

Operating System Device Handlers

Article on page 11

```
0100 ; MAKEBOOT         (FOR OS/A+)
0110 ; BY BOB MARTIN
0120 ; (c)1989, ANTIC PUBLISHING INC.
0130 ; MAKE BOOT DISK FROM DOS FILE
0140 ;
0150 MEMTOP = $6A
0160 MEMLO = $02E7
0170 HATABS = $031A
0180 CH   =   $02FC
0190 CR   =   $00
0200 BUF1 =   $0500
0210 BUF2 =   $0600
0220 ;
0230 INBUFF = $F3
0240 CIX =    $F2
0250 LBUFF = $0580
0260 FR0 =    $D4
0270 FR1 =    $E0
0280 ICBAL = $0344
0290 ICBLL = $0348
0300 ICAX1 = $034A
0310 ICAX2 = $034B
0320 CIOV =   $E456
0330 ICCOM = $0342
0340 AFP =    $D800
0350 FASC =   $D8E6
0360 IFP =    $D9AA
0370 FPI =    $D9D2
0380 ;
0390 ; LOOK FOR AN EMPTY SPOT IN
0400 ; THE HANDLER ADDRESS TABLE
0410 ;
0420      *=  $7000
0430 ;
0440 INSERT
0450      LDX #0
0460 NEXTPLACE
0470      LDA HATABS,X
0480      BEQ ESPOT
0490      INX
0500      INX
0510      INX
0520      BNE NEXTPLACE
0530      BEQ INSERT
0540 ;
0550 ;    FOUND AN EMPTY SPOT
0560 ;
0570 ESPOT
0580      LDA #$42
0590      STA HATABS,X
0600      LDA #BTAB&$FF
0610      STA HATABS+1,X
0620      LDA #BTAB/$0100
0630      STA HATABS+2,X
0640 ;
0650 ; FIND THE E: HANDLER
0660 ;
0670 EFIND
0680      LDX #0
0690 ELOOP
0700      LDA HATABS,X
0710      CMP #'E
0720      BEQ EFND
0730      INX
0740      INX
0750      INX
0760      BNE ELOOP
0770      BEQ EFIND
0780 ;
0790 ;    FOUND THE E: POINTER
0800 ;
0810 EFND
0820      LDA HATABS+1,X
0830      STA $D0
0840      LDA HATABS+2,X
0850      STA $D1
0860      LDY #4
0870      LDA ($D0),Y
```

```
0880      STA EGET+1
0890      INY
0900      LDA ($D0),Y
0910      STA EGET+2
0920      INY
0930      LDA ($D0),Y
0940      STA EPUT+1
0950      INY
0960      LDA ($D0),Y
0970      STA EPUT+2
0980 ;
0990      INC EGET+1
1000      BNE L013
1010      INC EGET+2
1020 L013
1030      INC EPUT+1
1040      BNE L012
1050      INC EPUT+2
1060 ; -----------------------------
4000 ; OS/A+ VERSION
4010 ;
4020 L012
4030 ;
4040 ;  SET THE LOMEM POINTER
4050 ;
4060 LMP
4070      LDA #LAST&$FF
4080      STA MEMLO
4090      LDA #LAST/$0100
4100      STA MEMLO+1
4110 ;
4120      LDA #IMSG&$FF
4130      LDY #IMSG/$0100
4140      JSR PRINTE
4150 ;
4160 CU20 RTS
4170 ;
4180 IMSG
4190      .BYTE CR,CR,"The device"
4200      .BYTE " Bn: is now"
4210      .BYTE " ready.",CR
4220      .BYTE "Use the copy command"
4230      .BYTE " to access.",CR
4240      .BYTE " (eg. COPY D:FILE,"
4250      .BYTE "B2: ",')+128
4260 ;
4270      *=  $2100
4990 ; -----------------------------
5000 ;
5005 ; HANDLER AND HANDLER TABLE
5010 ;  FOR "B:" (MAKEBOOT)
5015 ;  HANDLER
5020 ;
5025 ;HANDLER TABLE
5030 ;
5035 BTAB
5040      .WORD BOPEN-1
5045      .WORD BCLOSE-1
5050      .WORD NOFNT-1
5055      .WORD BPUT-1
5060      .WORD STATUS-1
5065      JMP NOFNT
5070 ;
5075 ;HANDLER VARIABLES
5080 ;
5085 HEADER .BYTE $06
5090 HOLDA  .BYTE $00
5095 HOLDX  .BYTE $00
5100 STATS  .BYTE $00
5105 SECNO  .WORD $01
5110 BYTNO  .BYTE $00
5115 MAXSEC .WORD $00
5120 INFOBY .BYTE $00
5125 MFLAG  .BYTE $00
5130 RUNADR .WORD $00 ;INTADR must
5135 INTADR .WORD $00 ;follow RUNADR
5140 LOADAD .WORD $00
5145 CURMEM .WORD $00 ;LASTAD must
5150 LASTAD .WORD $00 ;follow CURMEM
5155 ;
5160 ;HANDLER OPEN FUNCTION
5165 ;
5170 BOPEN
5175      LDA #$00     ;clear variables
5180      STA BYTNO
5185      STA SECNO+1
5190      STA MAXSEC
5195      STA MAXSEC+1
5200      STA MFLAG
5205      STA RUNADR
5210      STA RUNADR+1
5215      STA INTADR
```

```
5220      STA INTADR+1
5225      LDA #$01
5230      STA SECNO
5235      STA STATS
5240      LDA #$06
5245      STA HEADER
5250      JSR CLEAR
5255      CLC
5260      LDA $21       ;disk #
5265      ADC #$30      ;make ASCII #
5270      STA MSG2A
5275      LDA #MSG2&$FF
5280      LDY #MSG2/$0100
5285      JSR PRINTE
5290      JSR YESNO     ;Y/N?
5295      BNE L014
5300      LDA #$80      ;abort with
5305      STA STATS     ;error = 128
5310      JMP EXIT
5315 L014
5320      LDA #MSG1&$FF ;clear secs?
5325      LDY #MSG1/$0100
5330      JSR PRINTE
5335      JSR GETNUM
5340      BCS L014      ;error?
5345      LDA FR0
5350      STA BUF2
5355      LDA FR0+1
5360      STA BUF2+1
5365      ORA BUF2
5370      BEQ BOP04
5375 BOP01
5380      JSR PUTSEC    ;fill sectors
5385      INC SECNO     ;with zeros
5390      BNE BOP02
5395      INC SECNO+1
5400 BOP02
5405      LDA BUF2
5410      BNE BOP03
5415      DEC BUF2+1
5420 BOP03
5425      DEC BUF2
5430      BIT BUF2+1
5435      BPL BOP01
5440 BOP04
5445      LDA #0
5450      STA SECNO+1
5455      STA MAXSEC
5460      STA MAXSEC+1
5465      LDA #1
5470      STA SECNO
5475 ;
5480      JMP EXIT
5485 ;
5490 ;HANDLER SUBROUTINES
5495 ;
5500 ; 1) Store load address and
5505 ;    end of load address
5510 ;
5515 LODADR
5520      PHA           ;calc index &
5525      TYA           ;store load
5530      SEC           ;address &
5535      SBC #3        ;end of load
5540      EOR #$FF
5545      TAY
5550      INY
5555      PLA
5560      STA CURMEM,Y
5565      RTS
5570 ;
5575 ; 2) Get two 2-byte
5580 ;    load addresses
5585 ;
5590 LDINFO
5595      LDA INFOBY
5600      BNE L007
5605      LDA #4
5610      STA INFOBY
5615 L007
5620      DEC INFOBY
5625      LDY INFOBY
5630      LDA HOLDA
5635      JSR LODADR
5640      LDY INFOBY
5645      BEQ LD03
5650      CPY #2
5655      BNE LD02
5660 ;
5665      LDA CURMEM   ;check for
5670      CMP CURMEM+1 ;appended file
5675      BNE LD02
```

```
5680       CMP #$FF
5685       BNE LD02
5690       LDA #4
5695       STA INFOBY
5700 LD02
5705       JMP EXIT
5710 LD03
5715       JSR INRUAD
5720       BEQ LD02
5725       BNE L008
5730 ;
5735 ; 3> Check for load file run
5740 ;     or init. address
5745 ;     If CURMEM & LASTAD
5750 ;        are <$2E0 or >$2E3
5755 ;     THEN Z=0
5760 ;     ELSE Z=1
5765 ;
5770 INRUAD
5775       LDA CURMEM+1
5780       CMP LASTAD+1
5785       BNE IRA01
5790 IRA02
5795       CMP #2
5800       BNE IRA01
5805       LDA #$DF
5810       CMP CURMEM
5815       BCS IRA02
5820       CMP LASTAD
5825       BCS IRA02
5830       LDA #$E3
5835       CMP CURMEM
5840       BCC IRA02
5845       CMP LASTAD
5850       BCC IRA02
5855       LDA #0
5860 IRA01 RTS
5865       RTS
5870 ;
5875 ; 4> Check for load address
5880 ;     less than initial
5885 ;
5890 L008
5895       LDA LOADAD+1
5900       CMP CURMEM+1
5905       BCC LF01
5910       BNE LF02
5915       LDA LOADAD
5920       CMP CURMEM
5925       BEQ LF01
5930       BCC LF01
5935 LF02
5940       LDA #MSG6&$FF ;LOADAD>CURMEM
5945       LDY #MSG6/$0100
5950       JSR PRINTE
5955       LDY #168
5960       STY STATS
5965       JMP EXIT
5970 ;
5975 ; 5> Calculate sector and byte
5980 ;     for new load address
5985 ;
5990 LF01
5995       SEC            ;Find offset
6000       LDA CURMEM
6005       SBC LOADAD
6010       STA BUF2
6015       LDA CURMEM+1
6020       SBC LOADAD+1
6025       STA BUF2+1
6030       LDA BUF2       ;calc byte
6035       AND #$7F
6040       STA BYTNO
6045       LDX #7
6050 L009
6055       CLC            ;calc sector
6060       ROR BUF2+1     ;divide by 128
6065       ROR BUF2
6070       DEX
6075       BNE L009
6080       INC BUF2
6085       BNE LF11
6090       INC BUF2+1
6095 LF11
6100       LDA BUF2       ;Load sec if dif
6105       CMP SECNO
6110       BNE LDI1
6115       LDA BUF2+1
6120       CMP SECNO+1
6125       BEQ LDI2
6130 LDI1
6135       LDA BUF2
6140       STA SECNO
6145       LDA BUF2+1
6150       STA SECNO+1
6155       JMP GETSEC
6160 LDI2
6165       JMP EXIT
6170 ;
6175 ; 6> Save run and init. address
6180 ;
6185 STIRAD
6190       JSR INRUAD
6195       BNE IR01
6200       JSR CMPMEM
6205       BCC IR01
6210       LDA CURMEM
6215       AND #$0F
6220       TAY
6225       LDA HOLDA
6230       STA RUNADR,Y
6235       INC CURMEM
6240       JMP EXIT
6245 IR01
6250       JMP L005
6255 ;
6260 ; 7> Process header information
6265 ;    for start of load file
6270 ;
6275 FSTSIX
6280       CPY #$04
6285       BEQ FS51
6290       JMP FS01
6295 ;
6300 FS51
6305       LDA #FM1&$FF ;  specify
6310       LDY #FM1/$0100 ;load
6315       JSR PRINTE  ;   information
6320       JSR YESNO
6325       BNE FS02
6330       JMP FS01
6335 ;
6340 FS02
6345       LDA #FM2&$FF ; get sector
6350       LDY #FM2/$0100 ; count
6355       JSR PRINTE
6360       JSR GETNUM
6365       BCS FS02
6370       LDA FR0
6375       BEQ FS03
6380       STA BUF1+1
6385 ;
6390 FS03
6395       LDA #FM3&$FF ;  get load
6400       LDY #FM3/$0100 ;address
6405       JSR PRINTE
6410       JSR GETNUM
6415       BCS FS03
6420       LDA FR0
6425       ORA FR0+1
6430       BEQ FS03
6435       LDA FR0
6440       STA BUF1+2
6445       STA LOADAD
6450       LDA FR0+1
6455       STA BUF1+3
6460       STA LOADAD+1
6465 ;
6470 FS04
6475       LDA #FI4&$FF ;  get init
6480       LDY #FI4/$0100 ;address
6485       JSR PRINTE
6490       JSR GETNUM
6495       BCS FS04
6500       LDA FR0
6505       ORA FR0+1
6510       BEQ FS05
6515       LDA FR0
6520       STA BUF1+4
6525       LDA FR0+1
6530       STA BUF1+5
6535 ;
6540 FS05
6545       LDA #FM4&$FF ;  get run
6550       LDY #FM4/$0100 ;address
6555       JSR PRINTE
6560       JSR GETNUM
6565       BCS FS05
6570       LDA FR0
6575       ORA FR0+1
6580       BEQ FS06
6585       LDA #$4C
6590       STA BUF1+6
6595       LDA FR0
```

```
6600      STA BUF1+7
6605      LDA FR0+1
6610      STA BUF1+8
6615 ;
6620 FS06
6625      LDA #0
6630      STA HEADER
6635      JSR PUTSEC
6640      JMP LDINFO
6645 ;
6650 FS01
6655      LDA HOLDA    ;  get load
6660      STA LOADAD+1 ;  address
6665      LDA CURMEM   ;  from file
6670      STA LOADAD
6675      LDA HOLDA
6680      DEC HEADER
6685      LDY HEADER
6690      CPY #4        ;Y<4
6695      BCS FS07      ;no
6700      JSR LODADR
6705      JMP EXIT
6710 FS07
6715      CMP #$FF      ;load file?
6720      BEQ EXIT
6725      LDA #MSG5&$FF
6730      LDY #MSG5/$0100
6735      JSR PRINTE
6740      LDY #168      ;error if not
6745      STY STATS     ;load file
6750 ;
6755 ;GET STATUS AND EXIT HANDLER
6760 ;
6765 STATUS
6770 EXIT
6775      LDY STATS
6780      TYA           ;Error in Y
6785 NOFNT RTS
6790 ;
6795 ; 8) Clear output buffer
6800 ;
6805 CLEAR
6810      LDX #$7F
6815      LDA #$00
6820 CLR1
6825      STA BUF1,X
6830      DEX
6835      BPL CLR1
6840      RTS
6845 ;
6850 ; 9) Check for end of load
6855 ;    C=0 CURMEM  > LASTAD
6860 ;    C=1 CURMEM <= LASTAD
6865 ;
6870 CMPMEM
6875      LDA LASTAD+1
6880      CMP CURMEM+1
6885      BCC L003
6890      BNE L003
6895      LDA LASTAD
6900      CMP CURMEM
6905 L003 RTS
6910 ;
6915 ;HANDLER PUT BYTE FUNCTION
6920 ;
6925 BPUT
6930      STA HOLDA     ;save byte
6935      LDY HEADER    ;from CIO
6940      BEQ BP02
6945      JMP FSTSIX
6950 BP02
6955      LDA INFOBY
6960      BNE INF11
6965      JSR CMPMEM    ;end of load?
6970      BCS BP01      ;no
6975 INF11
6980      JMP LDINFO
6985 BP01
6990      JMP STIRAD    ;run or init?
6995 L005
7000      LDX BYTNO
7005      LDA HOLDA
7010      STA BUF1,X    ;save in output
7015      INC CURMEM    ;buffer
7020      BNE L004
7025      INC CURMEM+1
7030 L004
7035      INC BYTNO     ;end of sector
7040      BPL L006      ;yes
7045      JSR PUTSEC    ;save sector
7050      INC SECNO
7055      BNE LP04
7060      INC SECNO+1
7065 LP04
7070      JSR GETSEC
7075      LDA #0
7080      STA BYTNO
7085      BEQ EXIT
7090 ;
7095 L006
7100      JSR CMPMEM
7105      BCS EXIT
7110      JMP PUTSEC
7115 ;
7120 ;HANDLER CLOSE FUNCTION
7125 ;
7130 BCLOSE
7135      BIT STATS
7140      BPL BCL01
7145      JMP EXIT
7150 BCL01
7155      JSR PUTSEC
7160 ;
7165      LDA #1
7170      STA SECNO
7175      LDA #0
7180      STA SECNO+1
7185      JSR GETSEC
7190 ;
7195      LDA #MSG4&$FF ;   sector
7200      LDY #MSG4/$0100 ; count
7205      JSR PRINTE
7210      LDA MAXSEC
7215      STA FR0
7220      LDA MAXSEC+1
7225      STA FR0+1
7230      JSR PNUM
7235      JSR BCLSUB
7240      BEQ BCL03
7245      LDA MAXSEC
7250      STA BUF1+1
7255 ;
7260 BCL03
7265      LDA INTADR   ;   init.
7270      ORA INTADR+1 ;   address
7275      BEQ BCL04
7280      LDA #FM5&$FF
7285      LDY #FM5/$0100
7290      JSR PRINTE
7295      LDA INTADR
7300      STA FR0
7305      LDA INTADR+1
7310      STA FR0+1
7315      JSR PNUM
7320      JSR BCLSUB
7325      BEQ BCL04
7330      LDA INTADR
7335      STA BUF1+4
7340      LDA INTADR+1
7345      STA BUF1+5
7350 ;
7355 BCL04
7360      LDA RUNADR   ;    run
7365      ORA RUNADR+1 ;    address
7370      BEQ BCL05
7375      LDA #FM6&$FF
7380      LDY #FM6/$0100
7385      JSR PRINTE
7390      LDA RUNADR
7395      STA FR0
7400      LDA RUNADR+1
7405      STA FR0+1
7410      JSR PNUM
7415      JSR BCLSUB
7420      BEQ BCL05
7425      LDA #$4C
7430      STA BUF1+6
7435      LDA RUNADR
7440      STA BUF1+7
7445      LDA RUNADR+1
7450      STA BUF1+8
7455 ;
7460 BCL05
7465      JMP PUTSEC
7470 ;
7475 BCLSUB
7480      LDA #FM7&$FF
7485      LDY #FM7/$0100
7490      JSR PRINTE
7495      JMP YESNO     ;Y/N?
7500 ;
7505 ;I/O SUBROUTINES
7510 ;
7515 ; 1) Read and write to disk
```

```
7520 ;
7525 GETSEC
7530      LDY #$52
7535      LDA #$40
7540      BNE DISKIO
7545 PUTSEC
7550      LDA MAXSEC+1 ; Check for
7555      BEQ L010     ;  too many
7560      LDA MFLAG    ;  load sectors
7565      BNE L010
7570      LDA #MSG3&$FF
7575      LDY #MSG3/$0100
7580      JSR PRINTE
7585      INC MFLAG
7590 L010
7595      LDA MAXSEC+1
7600      CMP SECNO+1
7605      BCC PU02
7610      BNE PU01
7615      LDA SECNO
7620      CMP MAXSEC
7625      BCC PU01
7630 PU02
7635      LDA SECNO
7640      STA MAXSEC
7645      LDA SECNO+1
7650      STA MAXSEC+1
7655 PU01
7660      LDY #$57     ;  Use #$50
7665      LDA #$80     ;  for no-verify
7670 DISKIO
7675      STY $0302
7680      STA $0303
7685      LDA #$31
7690      STA $0300
7695      LDA $21
7700      STA $0301
7705      LDA #$80
7710      STA $0308
7715      LDA #$0F
7720      STA $0306
7725      LDA #BUF1&$FF
7730      STA $0304
7735      LDA #0
7740      STA $0309
7745      LDA #BUF1/$0100
7750      STA $0305
7755      LDA SECNO
7760      STA $030A
7765      LDA SECNO+1
7770      STA $030B
7775      JSR $E459
7780      BMI ERROR
7785      RTS
7790 ERROR
7795      LDA $0303
7800      STA STATS
7805      TAY
7810      RTS
7815 ;
7820 ; 2) Y/N - Result in accumulator
7825 ;      Y = 1
7830 ;      N = 0
7835 ;
7840 YESNO
7845      LDA #$FF
7850      STA CH
7855 YN1
7860      LDA CH         ;Get 1 character
7865      CMP #43        ;Y?
7870      BNE YN2
7875      LDA #1
7880      BNE YN4
7885 YN2
7890      CMP #35        ;N?
7895      BNE YN1
7900      LDA #0
7905 YN4
7910      PHA
7915      LDA #$FF
7920      STA CH
7925      PLA
7930      RTS
7935 ;
7940 ; 3) Print the integer in FR0
7945 ;
7950 PNUM
7955      JSR IFP
7960      JSR FASC
7965      LDA INBUFF
7970      LDY INBUFF+1
7975      JMP PRINTE
7980 ;
7985 ; 4) Get a number
7990 ;    result in FR0
7995 ;
8000 GETNUM
8005      LDA #$30
8010      STA LBUFF
8015      LDX #1
8020      STX HOLDX
8025 NOTCR
8030      JSR EGET      ;Get a number
8035      LDX HOLDX
8040      INC HOLDX
8045      STA LBUFF,X ;Store it at
8050      CMP #$9B      ;$580
8055      BNE NOTCR
8060      LDA #LBUFF&$FF ;Point to
8065      STA INBUFF   ;  ASCII string
8070      LDA #LBUFF/$0100
8075      STA INBUFF+1
8080      LDA #0
8085      STA CIX
8090      JSR AFP       ;ASCII to FP
8095      BCS GN01      ;error if C set
8100      JSR FPI       ;FP to integer
8105 GN01 RTS
8110 ;
8115 ; 5) Print text to screen
8120 ;    Low byte of text in
8125 ;    accumulator, High byte
8130 ;    of text in Y register
8135 ;
8140 PRINTE
8145      STA MSGN+1
8150      STY MSGN+2
8155      JMP MSGN
8160 EP02
8165      INC MSGN+1
8170      BNE MSGN
8175      INC MSGN+2
8180 MSGN
8185      LDA MSGN
8190      BEQ EP04
8195      BMI EP05
8200      JSR EPUT
8205      JMP EP02
8210 EP04
8215      JSR PNTCR
8220      JMP EP02
8225 ;
8230 EP05
8235      AND #$7F
8240      BEQ EP06
8245      JSR EPUT
8250 ;
8255 PNTCR
8260      LDA #$9B
8265      JSR EPUT
8270 EP06 RTS
8275 ;
8280 ;JUMP TO SCREEN EDITOR
8285 ; PUT BYTE ROUTINE
8290 ;
8295 EPUT JMP EPUT
8300 ;
8305 ;JUMP TO SCREEN EDITOR
8310 ; GET BYTE ROUTINE
8315 ;
8320 EGET JMP EGET
8325 ;
8330 ;
8335 FM1  .BYTE CR,"Do you wish to "
8340      .BYTE "specify the boot",CR
8345      .BYTE " sector information "
8350      .BYTE "(Y/N)",'?+128
8355 ;
8360 FM2  .BYTE CR,"Enter boot sector"
8365      .BYTE " count ",128
8370 ;
8375 FM3  .BYTE CR,"Enter load "
8380      .BYTE "address ",128
8385 ;
8390 FM4  .BYTE CR,"Enter run "
8395      .BYTE "address ",128
8400 ;
8405 FI4  .BYTE CR,"Enter "
8410      .BYTE "initialization "
8415      .BYTE "address ",128
8420 ;
8425 FM5  .BYTE CR,"Load file "
8430      .BYTE "initiation address "
8435      .BYTE "= ",128
```

```
8440 ;
8445 FM6  .BYTE CR,"Load file "
8450      .BYTE "run address "
8455      .BYTE "= ",128
8460 ;
8465 FM7  .BYTE CR,"Insert into "
8470      .BYTE "boot sector (Y/N)"
8475      .BYTE '?+128
8480 ;
8485 MSG1 .BYTE CR,"How many sectors"
8490      .BYTE " to clear",'?+128
8495 ;
8500 MSG2 .BYTE CR,"Make disk #"
8505 ;
8510 MSG2A .BYTE $31," into boot"
8515      .BYTE " disk (Y/N)",'?+128
8520 ;
8525 MSG3 .BYTE CR,"Sector count"
8530      .BYTE " exceeds 25",'5+128
8535 ;
8540 MSG4 .BYTE CR,"The boot sector"
8545      .BYTE " count is ",128
8550 ;
8555 MSG5 .BYTE CR,"Not a load"
8560      .BYTE " fil",'e+128
8565 ;
8570 MSG6 .BYTE CR,"Specified load"
8575      .BYTE " address is",CR
8580      .BYTE "  is not lowest"
8585      .BYTE " addres of fil"
8590      .BYTE 'e+128
8595 ;
8600 LAST .END
```

Tech Tips

GOTOLIST

BY GREGG HESLING

Don't type the TYPO II Codes!

```
DF 10 REM GOTOLIST
AA 20 REM BY GREGG HESLING
QH 30 REM (c)1988, ANTIC PUBLISHING
GF 31990 CLR :A=INT((FRE(0)-1000)/19)*19:
   DIM A$(A),B$(7):A$=" ":A$(A)=A$:A$(2)=
   A$:A=1:B=PEEK(136)+PEEK(137)*256
BP 31991 LINE=PEEK(B)+PEEK(B+1)*256:C=1:I
   F LINE=31990 THEN ? A$:END
LQ 31992 IF C=PEEK(B+2) THEN B=B+PEEK(B+2
   ):GOTO 31991
MK 31993 PK0=PEEK(B+C):PK1=PEEK(B+C+1):IF
    (PK1<>14 AND PK1<128) OR (PK0=27 AND
   PK1>=128) THEN 31999
JV 31994 RESTORE :FOR E=1 TO 8:READ D,B$:
   IF PK0<>D THEN NEXT E:GOTO 31999
HS 31995 IF PK0=30 THEN D=PEEK(B+C+2):D=(
   D=23 OR D=24)*2:C=C+D:IF D=0 THEN 3199
   9
KS 31996 PK1=PEEK(B+C+1):PK2=PEEK(B+C+2)-
   61:IF PK1>=128 OR PK2<0 THEN A$(A)="VA
   R  ":GOTO 31998
CN 31997 G=0:FOR D=3 TO PK2:E=PEEK(B+C+D)
   :F=INT(E/16):G=G+((E+F*10-F*16)*INT(10
   0^(PK2-D)+0.5)):NEXT D:A$(A)=STR$(G)
MC 31998 A$(A+6)=B$:A$(A+14)=STR$(LINE):A
   =A+19:C=C+2+6*(PK1<128):IF PEEK(B+C)=1
   8 THEN 31996
VG 31999 C=C+1:GOTO 31992:DATA 4,LIST,10,
   GOTO,11,GO TO,12,GOSUB,13,TRAP,27,THEN
   ,30,ON,35,RESTORE
```

Recently I purchased an unprotected BASIC program and found it was mostly "spaghetti code" — a mindless collection of GOTO, GOSUB, and TRAP statements! I immediately decided to re-write it, but it was too long and too complicated.

I was deathly afraid to change any lines, fearing another line would try to call it! Because it was a BBS program, I couldn't have some user get a READY prompt while I wasn't home!

In my typical fashion of trying to write programs which I understand nothing about, I seem to have succeeded with GOTOLIST.

GOTOLIST will search any BASIC program and find all line numbers referenced by other program lines. For example, GOTOLIST would save line 10 if it is:

10 GOTO 100

Although GOTOLIST is small, it's not stupid. If it encounters the following line:

10 GRAPHICS 0:? "HELLO.";:INPUT A$:IF A$ = "NO" THEN ? "OK":GOTO 100

all it saves is: 10 GOTO 100.

Type in the listing below, GOTOLIST.LST, check it with TYPO II, and *list* a copy to disk with the command:

LIST "D:GOTOLIST.LST",31990,31999

Now type NEW and LOAD a copy of that favorite program you've been dying to modify but are afraid of destroying. (Never modify your original copy—you're just asking for trouble!)

First, make sure your program doesn't use any line numbers above 31990. Then ENTER "D:GOTOLIST.LST" [RETURN] and type GOTO 31990. In a short while (It takes about a minute to check your program, so bring a good book if your program is long) GOTOLIST will display its list of lines which call other program lines.

If you want to see the list again, type PRINT A$, or LPRINT A$ for a printout. For best results, put your printer in 76-column mode by typing:

LPRINT CHR$(27);CHR$(81);CHR$(76)

Your printer may require a different code—consult your printer manual.

Now, before you change any line, just a glance at the list will insure nothing is trying to use that line for other purposes! If something is, however, just one more glance and you'll see which line is the culprit, and you can modify to your heart's content!

***Antic** pays $25 for every original and exclusive Tech Tip submission that we publish. Send your 8-bit or ST disk and printout to: Antic Tech Tips, 544 Second Street, San Francisco, CA 94107. Tech Tips welcomes very short programs that demonstrate the Atari's powers, simple hardware modifications, or useful macros for popular software.*

Antic Classifieds

Advertisers Index

This list is provided as a courtesy to our advertisers. ANTIC does not guarantee accuracy or comprehensiveness.

Advertising Deadlines

September 1989

Insertion Orders: May 31
Artwork: June 7
On Sale: Last week of July

October 1989

Insertion Orders: June 30
Artwork: July 7
On Sale: Last week of August

November 1989

Insertion Orders: August 1
Artwork: August 8
On Sale: Last week of September

Missing any issues? Looking for those great utilities, games, and reviews? You'll find it all right here! Copies are limited, so order today!

ORDER #	ISSUE
1	**April 1983:** Games—3-D Maze, Voyeur (No Disk)
2	**May 1983:** Telecomputing—Microids, Tele Chess (No Disk)
3	**June 1983:** Databases—Stargazing, Dancin' Man (No Disk)
4	**July 1983:** Adventure Games—Dragonsmoke, Shoot-em-up Math
5	**August 1983:** Graphics—3-D Fuji, Keystroke Artist
6	**September 1983:** Education—P/M Tutor
7	**October 1983:** Sports Games—AutoCassette
8	**November 1983:** Sound & Music—Air Raid, Casting Characters
9	**December 1983:** Buyer's Guide—AUTORUN.SYS, Automate Player/Missiles
10	**January 1984:** Printers—Pocket Calendars, Screen Dump
11	**February 1984:** Personal Finance—TYPO, Gauntlet
12	**March 1984:** International Issue—DiskRead, Poker Solitaire
13	**April 1984:** Games—Risky Rescue, Math Wizard
14	**May/June 1984:** Exploring XL Computers—Escape from Epsilon, Scroll to the Top
15	**July 1984:** Communications—AMODEM, BASIC Animation Secrets
16	**August 1984:** Disk Drives—Horseplay, Recall
17	**September 1984:** Computer Graphics—Graphics Converter, Olympic Dash (No Magazine)
18	**October 1984:** Computer Learning—Bouncing Ball, Antic 4/5 Editor/Animator
19	**November 1984:** Computer Adventures—Adventure Island, Advent X-5
20	**December 1984:** Buyer's Guide—Infobits, Biffdrop
21	**January 1985:** Super Utilities—TYPO II, DISKIO
22	**February 1985:** Finances—Home Loan Analyzer, Drum/Bass Synth
23	**March 1985:** Printers—Kwik Dump, Font Maker
24	**April 1985:** Computer Frontiers—Dot Matrix Digitizer, Speech Editor
25	**May 1985:** New Super Ataris—Son of Infobits, Arena Racer
26	**June 1985:** Computer Arts—View 3-D, The Musician
27	**July 1985:** Computer Challenges—Miniature Golf, Guess That Song
28	**August 1985:** Telecommunications—Atari 'Toons, Pro*Term
29	**September 1985:** Power Programming—One-Pass Disk Copy 130, Crickets

ORDER #	ISSUE
30	**October 1985:** Mind Tools—Graph 3D, GEM Color Cascade
31	**November 1985:** New Communications—TYPO II Double Feature, 130XE Memory Management
32	**December 1985:** Shoppers Guide—DISKIO Plus, Box-In
33	**January 1986:** Atari Products are Back—Appointment Calendar, Dungeon Master's Apprentice
34	**February 1986:** Printer Power—T-Shirt Construction Set, Forth Escapes
35	**March 1986:** Practical Applications—Lunar Lander Constructor, Lie Detector
36	**April 1986:** Computer Mathematics—Fractal Zoom, 3-D Fractals
37	**May 1986:** 4th Anniversary—Digital Gardener, Molecular Weight Calculator
38	**June 1986:** Summer Computing—Weather Wizard, Bomb Squad
39	**July 1986:** Computer Arts—Amazing Card Shuffler, Grafcon ST
40	**August 1986:** Online Communications—Ultrafont, Floppy Filer
41	**September 1986:** Weather—WEFAX Decoder (8-bit/ST), BASIC Tracer
42	**October 1986:** Hard Disks—Video Stretch, TYPO ST
43	**November 1986:** Personal Finance—Budget dataBASE, V-Graph
44	**December 1986:** Shoppers Guide—Stepper Motors, Nuclear Waste Dump
45	**January 1987:** Talking Atari—Talking Typewriter, Rebound
46	**February1987:** Word Processing—SF Fogger, Electric Charlie!
47	**March 1987:** Dvorak Keyboard, Multi-AUTORUN
48	**April 1987:** —Designer Labels, Taxman
49	**May 1987:** 5th Anniversary—A-Rogue, Poker Slot Machine
50	**June 1987:** Animation—Verbot Commander, Citadel
51	**July 1987:** Print Anything—Ghost Writer, Your Net Worth
52	**August 1987:** Atari Muscle—Sideways Spreadsheet, Diamond Dave
53	**September 1987:** Work/Play—Mighty Mailer, Maximillian B.
54	**October 1987:** Football Predictor—Antic Prompter (8-bit/ST), Spelling Checker
55	**November 1987:** Practical Applications—Critical-Path Projects, WYSIWYG Cassettes
56	**December 1987:** Print Holiday—P.S. Envelope Maker, Antic Publisher

TERRIFIC SOFTWARE PRESENTS

TWO NEW, SENSUOUS GAME TITLES

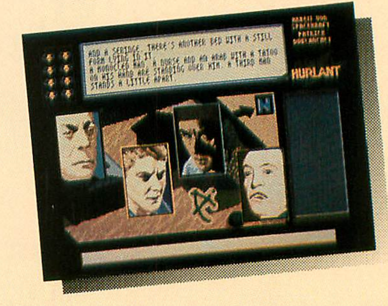
Look out, Indiana Jones! Step aside, Bogie! *Crash Garrett's* in town!

Don't wait 'till Sunday to catch-up on your favorite action comic — play ***CRASH GARRETT*** instead!

No other adventure game is quite like this…







Let ace flyer ***CRASH GARRETT*** escort you through Hollywood in the '30s to rescue sultry, sexy gossip columnist, Cynthia Sleeze, from the sinister Nazi mastermind Baron von Engel Krul and his cronies. Help CRASH stop this perverse Nazi spy-ring from kid-napping glamorous, American beauties to use as breeding stock for an Aryan race of superhumans. Be the voice in CRASH's head as he encounters adventure after adventure with a whole group of wacky, depraved characters including Caleb Thorn, psychoanalyst to the stars, and Lotta Linebacker, a female wrestler who knows what she wants from a man!

CRASH GARRETT is style and pizzazz—an animated comic book with a slick, continental look. It's about as much terror, intrigue and suspense you'll want from any game!

Play *Stir Crazy With Bobo*—

Your idea of "doing time" will definitely change!

Had a little too much violence lately? Still want fun and action? Well, grab your joystick and join poor Bobo in six of the most graphically amusing adventure games ever on disk.



Bobo's in prison—INZEESLAMMER—where he spends most of his time performing menial chores and planning his escape. Bobo's *no penal pushover!*

Bobo starts his day feeding porridge to hungry, irritable prisoners. Don't let him get too befuddled, or else he'll end up with the porridge bucket on his head.



K.P.'s next. Speed is the key here. Don't let Bobo get buried underneath a pile of spuds!

An exhausted Bobo tries to catch up on his beauty sleep, but is constantly interrupted by the relentless snoring of his cell mates.

Finally Bobo makes his escape and hurtles off into the sunset, right onto electric train cables. Bobo needs some pretty fancy footwork to avoid the pulsating current!

